

# How Do You Do This with ColdFusion 9 ORM ?

Posted At : November 25, 2009 6:35 PM | Posted By : Jon Hartmann

Related Categories: ColdFusion

OK, so I've been working through build a new blog platform using the new ColdFusion 9 ORM setup, and I'm really impressed with how much of the basics it handles for you, but I'm starting to get into stuff thats more than just a simple CRUD action, and I'm finding that I've no idea how to make the ORM or HQL get me the results I need. Perhaps you can help me figure out what I need to do.

## The Setup

The database I'm working with has basically three elements: Posts, Comments, and Categories. These three tables are mapped as objects with names Post, Comment, and Category, and have a very classic relationship structure; Comments have a many-to-one relationship to Posts, and Posts have a many-to-many relationship with Categories through a table called PostCategories.

## How Many Comments Does this Post Have?

This might be the simplest of my questions, but how do you get the count of how many comments a post has in an efficient manner? The obvious answer is:

```
ArrayLen(post.getComments())
```

Which is all fine and good, but you're looking at getting all of the data from all comments just to find out how many there are aren't you? If we switch to ORMExecuteQuery() with HQL we can do:

```
ORMExecuteQuery("SELECT COUNT(*) FROM Comment WHERE postID = ?", [post.getPostID()])
```

Or some variation there on, and we can get back the count, but thats really not any better that just writing the SQL ourselves, and there isn't really an easy way to abstract this, so we'd have to do what, write a custom method on every ORM object to get this back?

## To How Many Categories Does This Post Belong?

Given a post to how many categories does it belong? This is very similar to the last problem, accept that I've no idea how make the HQL even work for it. I know that I can do:

```
ArrayLen(post.getCategories())
```

And that that will get me the count, but it wont be very efficient since I'm getting all that category info, just like the comments. The problem is now that I can't just write the HQL, because the HQL doesn't really know about PostCategories, at least as far as I can tell. I tried writing:

```
ORMExecuteQuery("SELECT COUNT(*) FROM PostCategories WHERE postID = ?", [post.getPostID()])
```

And all I got was an error. Perhaps I need to define PostCategories as a persistent object for this to work? That doesn't seem right, since I thought that the whole relationship thing existed to hide this complexity.

## How do I get Posts 5-10 that are in Category X by Date?

Given a category, how do you find a particular range of posts that belong to that category? I love the fact that the basic EntityLoad() and ORMExecuteQuery() functions are very pagination friendly, but that doesn't seem to be true for dealing with associated entities. EntityLoad() can't get me what I want, because it doesn't seem to understand if a given Post belongs to a given Category, since its not in the table, and ORMExecuteQuery() doesn't seem to have this either since it doesn't know that PostCategories exists. I can use category.getPosts() to get ALL the posts, but again, that doesn't seem very efficient at all. The date ordering complication throws another curve-ball in there; even assuming I don't mind getting back every post with category.getPosts(), if I need things in a different order than was set in the relationship, there doesn't seem to be a way to change that either.

## I have to be Missing Something

Given these questions, I feel like I'm missing something. I've gone over the ORM documentation a couple of times, but I can't find anything on HQL joining tables, filters for .get<Entity>() or anything that would get me closer to a solution, but it just feels like these have to be very commonly encountered issues. What is it that I'm missing?