

# JDBC Drivers, Data Binding and Implicit Conversion

Posted At : February 25, 2007 12:13 AM | Posted By : Mark Kruger

Related Categories: MS SQL Server, Coldfusion & Databases, Coldfusion Upgrading

You may know that MSSQL allows for "implicit conversion" between data types. For example if you have a character column and you pass it a number without single quotes (as in `myCharCol = 1`), MSSQL will automatically convert the value 1 into a character for the purpose of the query. On a Coldfusion 5 server this behavior carries through seamlessly even when you are binding data using `cfqueryparam`. However, if you are using `cfqueryparam` in a query on a CF 5 server and also relying on implicit conversion within the *same query* you should know that you may have a problem getting that `cfquery` to work in Coldfusion MX. The issue is that JDBC "prepares" the statement by validating against the schema. Here's an example:

Let's say we have a varchar column called "addDate" in a table that stores a string that contains a syntactically correct date. The following query would work in Coldfusion 5.

```
<cfquery name="myQuery" datasource="myDsn">
  SELECT      *
  FROM        users
  WHERE       user_id = <cfqueryparam cfsqltype="CF_SQL_INTEGER" value="#id#"/>
  AND         (DATEDIFF(dd,addDate,GETDATE()) >= 45)
</cfquery>
```

For the `DATEDIFF( )` call to work the column must either be a date datatype *or* it must contain a string that can be converted to a date datatype. The database drivers that ship with Coldfusion 5 do not use the same mechanism to check and see whether the column is a date datatype. So CF 5 hands off to the database server without "knowing" whether this function will work.

The JDBC drivers that ship with Coldfusion MX are a bit different. They work against a schema that is introspected from the database when the connection is first made (see my post on [schema caching](#) for a bit more information on this mechanism). When CFMX encounters the query above it knows the data type for "addDate" is actually a varchar. Since it does not support implicit conversion it throws a datatype mismatch error. Please note that this only happens when you are using `cfqueryparam` in the same query (as in our example) - because data binding causes the JDBC driver to prepare the statement before sending it to the db server.

## The Fix

The fix, obviously is not to rely on the implicit conversion facilities of MS SQL. Choose the datatype that makes sense for the functions and tasks that you need to accomplish - and learn to use the `CAST` and `CONVERT` functions when a change is necessary. If you are having a post migration problem like this a work around would be to remove the `cfqueryparam` tag. So, the example above rewritten like this...

```
<cfquery name="myQuery" datasource="myDsn">
  SELECT      *
  FROM        users
  WHERE       user_id = #id#
  AND         (DATEDIFF(dd,addDate,GETDATE()) >= 45)
</cfquery>
```

...would actually work on both platforms. Why? Because without `cfqueryparam`, JDBC

does not prepare the query in the same way. Instead it passes the query as a string to the database server and allows it to do all the work. Keep in mind that if you use this work around you are exposed to SQL injection attack and you will need to scrub all the variables you pass into the query. You also lose the other benefits of `cfqueryparam` - so I would consider this only a temporary fix until you find a way to alter the schema appropriately. You should be using `cfqueryparam` in virtually all cases (with only a few exceptions).

Special thanks to the regulars on the BACFUG email list for sparking this post.