# Removing Duplicate Records in a Database

Posted At : February 2, 2006 5:57 PM | Posted By : Mark Kruger
Related Categories: SQL tips

SQL provides many ways of grouping data. It also has many ways to select data. But if you've ever tried to remove duplicates from a database you might have needed 2 or 3 trips to the liquor store to figure it out. I've seen routines that match and compare and order and update and delete willy nilly - all just to find and fix duplicate rows in a database. I've never had to do this to any of *my* databases of course, because all my constraints and applications are perfect (gah!). Still, for those of you with imperfect databases I'm happy to report there *is a magic bullet*. Here's the big secret.

### Self Joining

Unless you are Narcissus it's not something you think about often, but it *is* possible to join a table with itself. This trick can be used to weed out duplicates - as in the query below.

```
<cfquery name="getDups" datsource="#mydsn#">
   SELECT    distinct a.user_id
   FROM    users a, users b
   WHERE    a.user_id > b.user_id
   AND    a.email = b.email
   ORDER BY a.user_id
</cfquery>
```

This little beauty would return all the rows with the same email that had been entered *after* the first record. So if I had 3 records:

| userid | email |
|---|---|
| 100 | myemail@myemail.com |
| 135 | myemail@myemail.com |
| 210 | myemail@myemail.com |

It would return the 135 and the 210 row. It would NOT return the 100 row.

You can easily use this to your advantage. For example, you could delete the duplicates from the table.

```
<cfquery name="getDups" datsource="#mydsn#">
   DELETE FROM users
   WHERE   user_id IN
      (SELECT    distinct a.user_id
      FROM    users a, users b
      WHERE    a.user_id > b.user_id
      AND    a.email = b.email)
</cfquery>
```

That's pretty nifty. It seems so simple, now that I look at it, I wonder why I've never seen it or thought of it before. I guess I just haven't been looking hard enough. I'm sure my readers will be happy to inform me (ha).