
Subject: MySql, boolean values and SqlArray
Posted by [Giorgio](#) on Wed, 27 Jan 2016 09:00:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi there,

MySql rdbms do not have a boolean value type. According to the current documentation BOOL or BOOLEAN "types are synonyms for TINYINT(1). A value of zero is considered false. Nonzero values are considered true" and that should be used to store boolean values. I noticed that this introduce a little problem. For instance, I have a table containing machinists IDs and a bunch of tinyint fields: if the field is "true" (i.e. nonzero value) the machinist can operate a certain machine. I tested it putting some values:

```
ID Name M1 M2 M3
'001' 'Titus' '1' '0' '2'
'002' 'Caius' '0' '1' '1'
'003' 'Sempronius' '1' '0' '0'
```

According to the documentation the following query should return 001 and 002: "Select ID, From MACHINISTS Where M3 = true;", but actually it returns only 002. If I use the following query: "Select ID, From MACHINISTS Where M3 is true;" I get correct results, i.e. 001 and 002. So MySql is picky, but is coherent.

Get back to Ultimate++ now. I have a SqlArray control that manages that table. I need to edit the M1, M2, M3 fields. So I used lines like the following:

```
EditInt m01abil, m02abil, m03abil;
[...]
tblMachinist.AddColumn(M1, "Abil. M01",10).Edit(m01abil);
```

And that's fine, but when I query the db:

```
query * Select(ID, NAME).From(MACHINISTS).Where(M! == true);
while (query.Fetch())
{
  Oper1.Set(i, (String)query[ID], (String)query[NAME]);
  i++;
}
```

I get just 002. So if the user puts 2 instead of 1 there are problem. To solve that I modified the code managing the SqlArray. Now it reads (note the type of variables):

```
Option m01abil, m02abil, m03abil;
[...]
```

```
tblMachinist.AddColumn(M1, "Abil. M01",10).Edit(m01abil);
```

I tried to declare in the .sch files the M1 fields both as BOOL and INT, but every time I have problem querying the db: if M3 is not 1 it is considered false.

Using an option type to me solved the issue, as users can not put values different from zero or one.

I do not know if ultimate++ can manage this kind of issue, but as I spent some time to figure this out I decided to share my solution.

Regards,
Giorgio
