
Subject: Re: PostgreSQL does not handle BOOL correctly [BUG+PATCH]

Posted by [mirek](#) on Sat, 21 Oct 2023 15:22:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

omari wrote on Sat, 21 October 2023 01:14 This test case (attached) FAIL without the patch, and PASS with the patch.

```
if(!OpenDB()) return;
```

```
Sql sql(session);  
sql * Insert(TEST1) (ID, 1) (B, true);  
sql * Insert(TEST1) (ID, 2) (B, false);
```

```
sql*Select(SqlAll()).From(TEST1).Where(B == true);  
LOG(sql.ToString());  
if (sql.Fetch()) {  
    ASSERT(sql[ID] == 1);  
}  
else {  
    ASSERT(false); // Failed : 'Select' should return one row  
}
```

```
sql*Select(SqlAll()).From(TEST1).Where(B == false);  
LOG(sql.ToString());  
if (sql.Fetch()) {  
    ASSERT(sql[ID] == 2);  
}  
else {  
    ASSERT(false); // Failed : 'Select' should return one row  
}
```

the schema file:

```
TABLE_(TEST1)  
INT_ (ID) PRIMARY_KEY  
BOOL_ (B)  
END_TABLE
```

That is all good and fine, just not the way it was intended to work. (And I do not claim that the intended way is the best but it worked fine for 20+ years.)

```
sql*Select(SqlAll()).From(TEST1).Where(B == '1')
```

is not all that harder to do and IMO it is more "honest" (we do not pretend that B is not char).

Frankly, the only difference between BOOL and STRING(1) apart from documentation purposes is that gets converted to bool in S_ structures. If you do not like that, do not use BOOL.

Mirek
