
Subject: Re: MSSQL error management
Posted by [mirek](#) on Sat, 24 Feb 2018 18:21:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Giorgio wrote on Fri, 16 February 2018 10:36Hi there,
to execute SQL queries I use code like this:

```
bool ImportData::InsertBody(myS_Table record)
{
    Sql sql(mssql);
    sql.ClearError();

    try { sql * Insert(My_Table)
        (My_Id, record.Id)
        (My_Row, record.Row)
        (My_Description, record.Description);
        return true;
    } catch(SqlExc) {
        ErrorOK(t_("Failed adding data to the database due to the following error: ") +
SQL.GetLastError());
        return false;
    }
    return true;
}
```

In this way, (1) if there are errors in executing the query the user is notified of the specific reason and (2) I can control the program flow using the true/false value returned by the method.

This is working as expected for MySql, PostgreSql and SQLite, but here it comes Microsoft and its @#!&\$ MS Sql Server.

When there is an error (e.g. a duplicate key) the user is NOT notified and the query returns always true. The only way to catch errors is looking at the log file activated using .LogErrors() and SetTrace(). This is not really user friendly. In case of an error what I got in the log file is the following:

```
ERROR [Microsoft][SQL Server Native Client 11.0][SQL Server]Violation of PRIMARY KEY
constraint 'Id'. Cannot insert duplicate key in object 'dbo.X_DORIG'. The duplicate key value is
(180107, 1, 'Art. number'                                ).(0): insert into X_DORIG(Id, Row, Description,)
values (180107, 1, 'Art. number')
```

I tried using both Driver={SQL Server Native Client 11.0}; and Driver={SQL Server}; with the same result.

Is there a workaround for this issue?

Thanks,
Gio

You do have exception active (SqlSession::ThrowOnError), right?

Mirek
