**Stored Procedures – DML Logic**

Stored Procedures can contain many different types of statements. These could include queries, inserts, updates, and deletes. A DML statement in a stored procedure is the same as if you wrote it outside the stored procedure.

There are some additional considerations when you use DML statements in stored procedures. We should keep in mind that they will often be called from client applications coded in VB.Net or ASP.NET (for example). If an error occurs in a DML statement in a stored procedure it would send a rather cryptic error message to the client application and possibly displayed to the user. These messages could include constraint error messages (check, foreign key, primary key, etc…) that you have probably seen before. Since those messages probably don’t mean much to the average user we would like to check to see if a DML statement failed and if it did send a more understandable message back to the client.

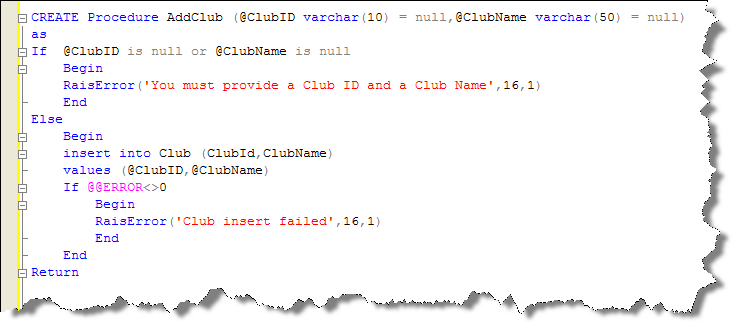
To accomplish this everY DML statement in our stored procedures will follow this structure:

DML statement

Check if it failed

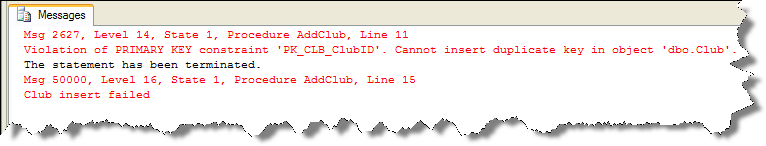
If it failed raise an error with a ‘nice’ error message

**Create a Stored Procedure to insert a new Club**



Remember, @@error maintains a value of 0 unless an error occurs. So, if @@error is not 0 an error in the most recent statement has occurred. We then use RaisError to send a message of our choosing to the client application.

Inserting a club with a ClubID that already exists causes a Primary Key constraint error.

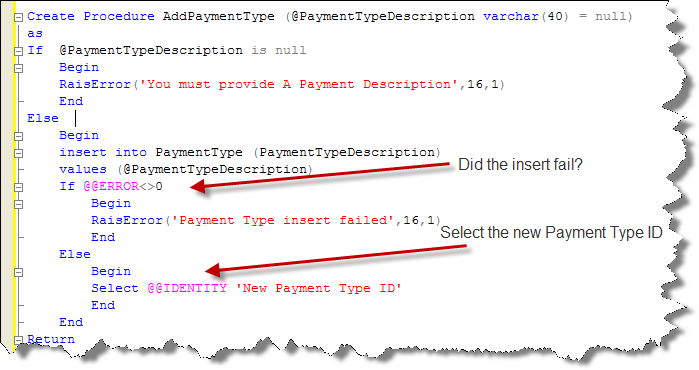


Note that the server still sends its own error message but we also send our own. The code on the client could be written to only show our RaisError message and suppress the servers more cryptic message.

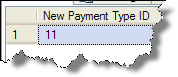
In this course we will see both in the messages window in management studio.

How could we use @@identity in a Stored Procedure?

Create a Stored Procedure to insert a new Payment Type and return the new PaymentTypeID.



**Result**

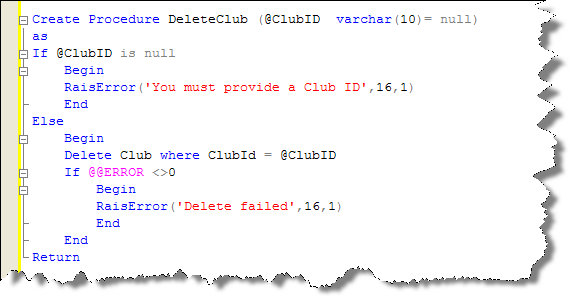


\*\***AFTER EVERY DML STATEMENT IN A STORED PROCEDURE WE WILL CHECK @@ERROR AND RAISERROR IF IT IS NOT 0 TO SEND A MESSAGE TO THE USER\***

**Updates and Deletes – Something to consider…**

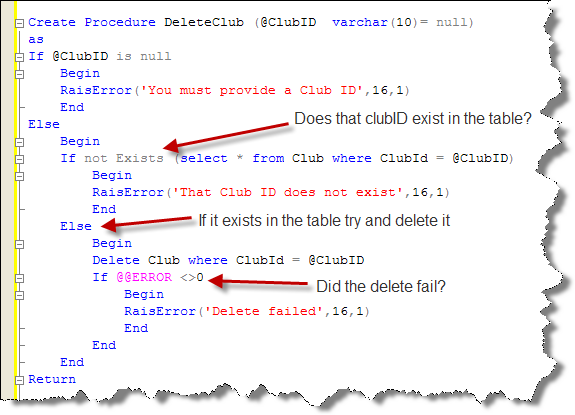
What happens when you try and update or delete a record that does not exist? Well, not much. The server simply says that 0 rows were affected.

This does, however, pose an interesting problem for users. Let’s say that the user supplies the stored procedure a club Id that they would like to delete. A stored procedure to accomplish this might look like this:



What would happen if the ClubID the user provides does not exist or they mistype it? Well, again, not much. The server would simply state 0 rows affected. The user would assume the delete succeeded as the server did not return an error. A better approach might be to check to see if the record exists that user is trying to delete and let them know it does not exist. While this might not always be the desired logic it can be accomplished with the exists clause.

This Stored Procedure checks for the existence of the record that the user is trying to delete. If the record is not there the user is notified by a RaisError.



In this course you will check for the existence of records that may not exist before you delete or update them. You will alert the user by a RaisError statement if the record does not exist.