# Automated Database Testing: Stored Procedures

Mary R. Sweeney

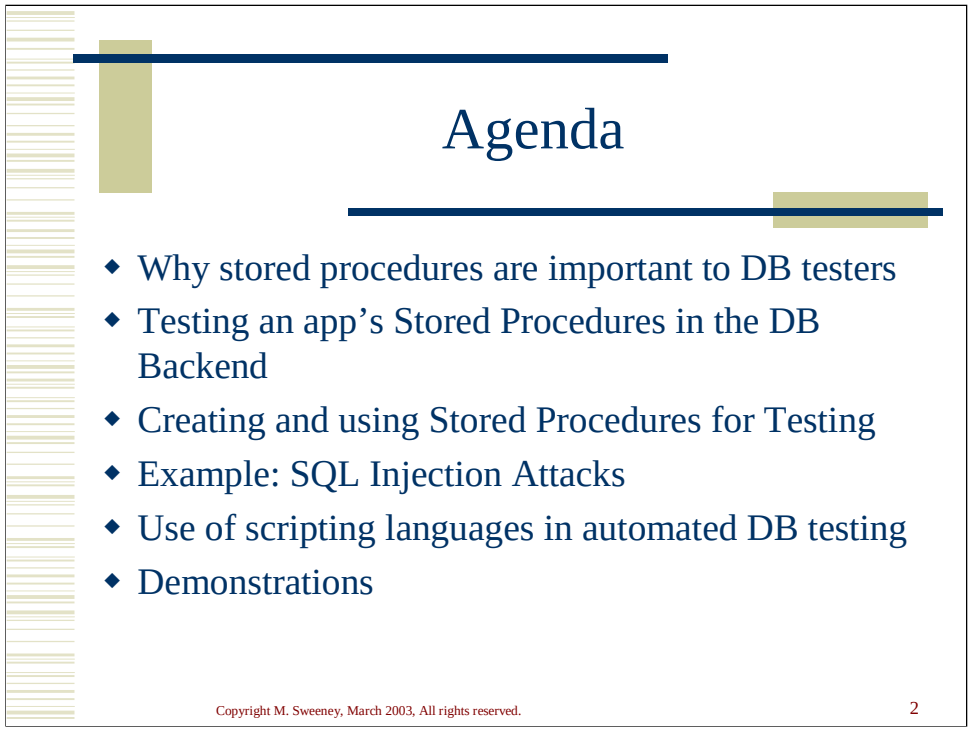
*Exceed Training*

www.exceedtraining.com





Today's heterogeneous data environments place an increasingly heavy burden on test engineers. Applications, whether web-based or client-server, must be tested for seamless interface with the backend databases; this typically goes far beyond what the popular test automation tools can provide. Testers are required to know how to create and use SQL, stored procedures, and other database objects to effectively test today's data driven environments. They also need to know how to successfully test objects such as stored procedures and views in an application's databases.




This presentation will cover tips and techniques for creating efficient automated tests of the critical database backend using simple scripting languages and relational database objects.

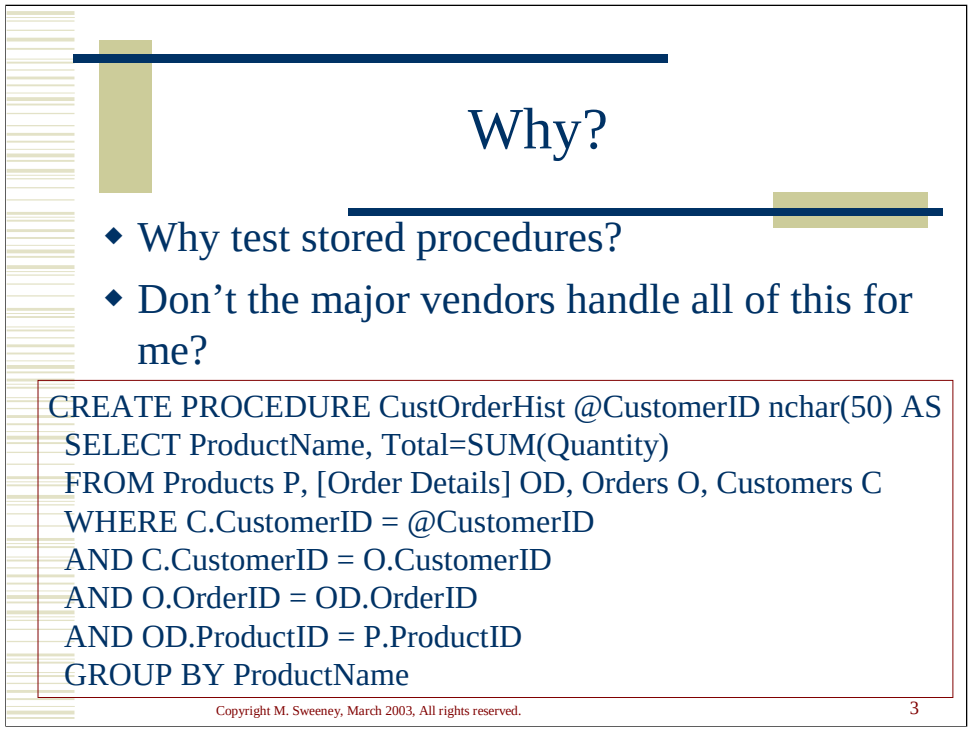
Attendees will learn

•Why testing of database objects and stored procedures is necessary and why popular automated tools can't keep up

•How to successfully test database objects such as stored procedures and views with examples and code

•Specific procedures, queries, views and other relational database objects that are valuable for typical testing situations

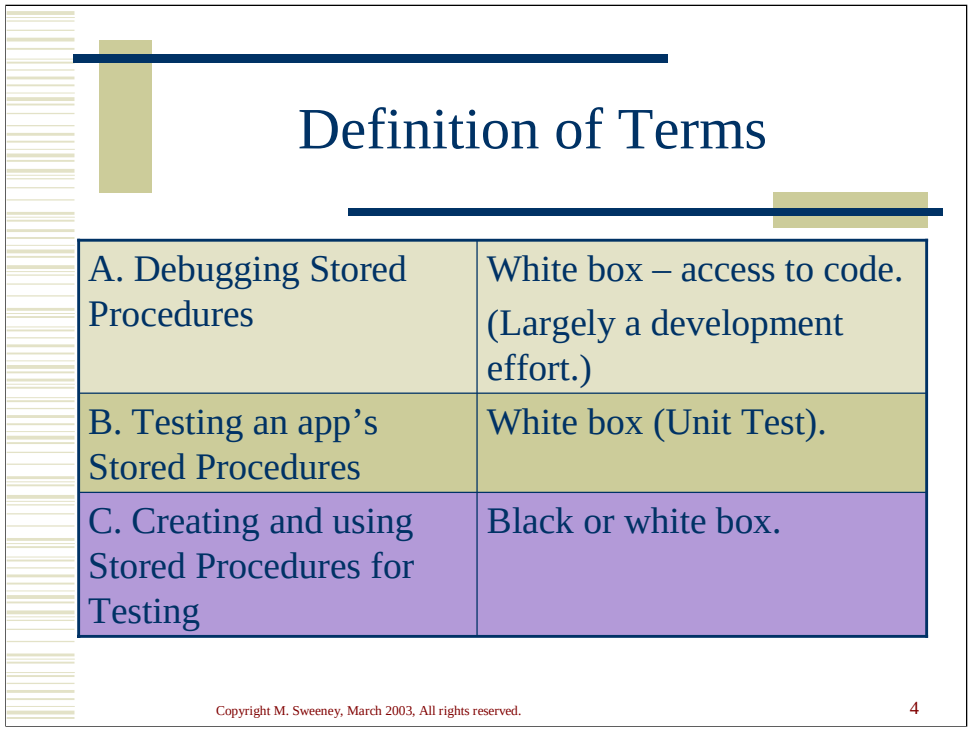
•How simple and effective automated tests for the backend can be created using programming languages including PERL and VBScript.




Since you're a tester, you always ask "why"? Why am I here? What's the point? That's what I love about this profession, because it's rife with curious people. I'm one of them. If you're wondering why you're here, let me give you a little rationale. First of all, Exceed Training's website receives many hits, we've noticed lately that the greater percentage of these hits are as a result of those who have gotten to use as a result of a search for "stored procedure tests" or similar queries. I've also found in my career throughout my career that the most valued skill I have, whether as a tester or a developer, involves database ability. How do companies ensure that correctness of their data? Corrupt data is worse than no data at all.

With the increasing use of data on the internet, and all of the associated headaches, the need for skilled testers in this area is growing quickly. Stored procedures are increasingly used by experienced database developers to ease the complexity of calls to the database backend. These procedures contain critical tasks such as inserting customer records and sales. They need to be tested at several levels. Black box testing of the front-end application is important, but makes it difficult to isolate the problem. Testing at the backend can increase the robustness of the data.

Doesn't my major vendor cover all of this? Not that I've seen! Sorry! I don't want to name names here but we all know who they are. They seem to have enough to do as it is providing the support for GUI testing and performance. Testing database objects can be done by pretty much all the tools in some limited fashion, but you don't necessarily have all the flexibility you need in the IDE of these tools and therefore have to rely on coding using their macro language anyway. As much as possible, I recommend bypassing their programming macro language and writing your tests in SQL script code directly in your RDBMS or using a variety of relatively simple scripting languages – VBScript, PERL, Ruby, Javascript. Why, you ask again? Because they're more reusable and portable in terms of your database, of course, and also because the skill set for using these languages will be more available in the personnel you're able to hire, train, and retain. I can think of other reasons too, but those are pretty good, don't you think?

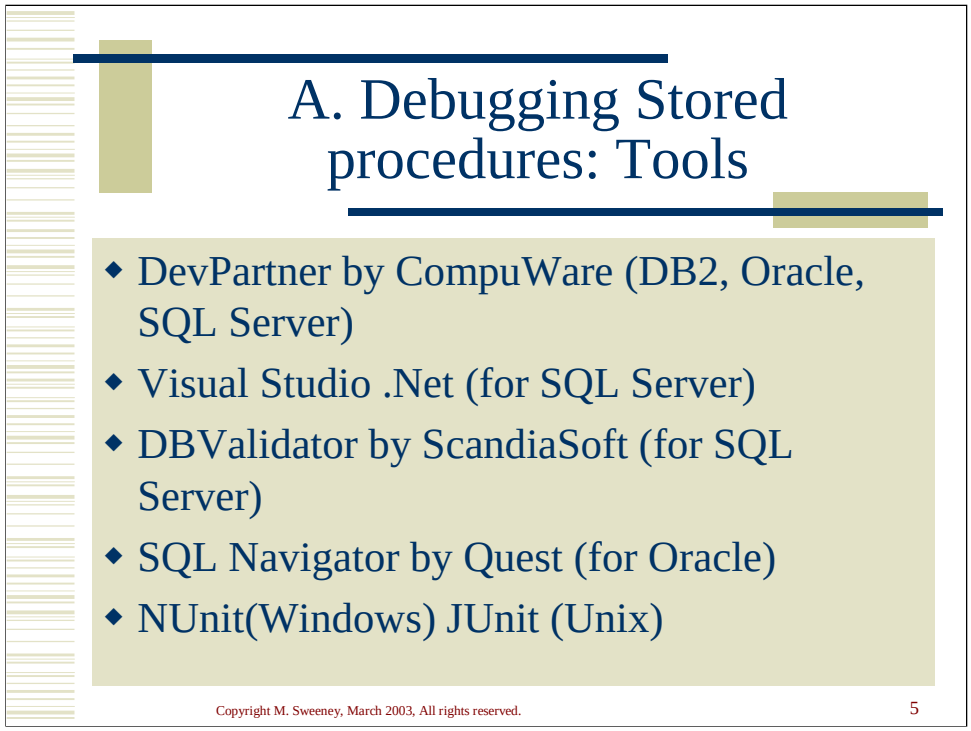



I could probably write separate papers on all three of these areas.

A. Many tools exist for specific debugging of stored procedures at the RDBMS level. For example, DB2, Oracle, SQL Server, all have their own utilities and strategies for debugging a stored procedure. This is mostly a development issue, but, of course, reflects on testing since developers rarely get all the bugs out! This presentation will not cover all the issues of debugging stored procedures but I will give you a list of tools that can support this effort.

B. Testing an app's stored procedures in the backend can be done is several ways, which we'll discuss specifically later in this presentation.

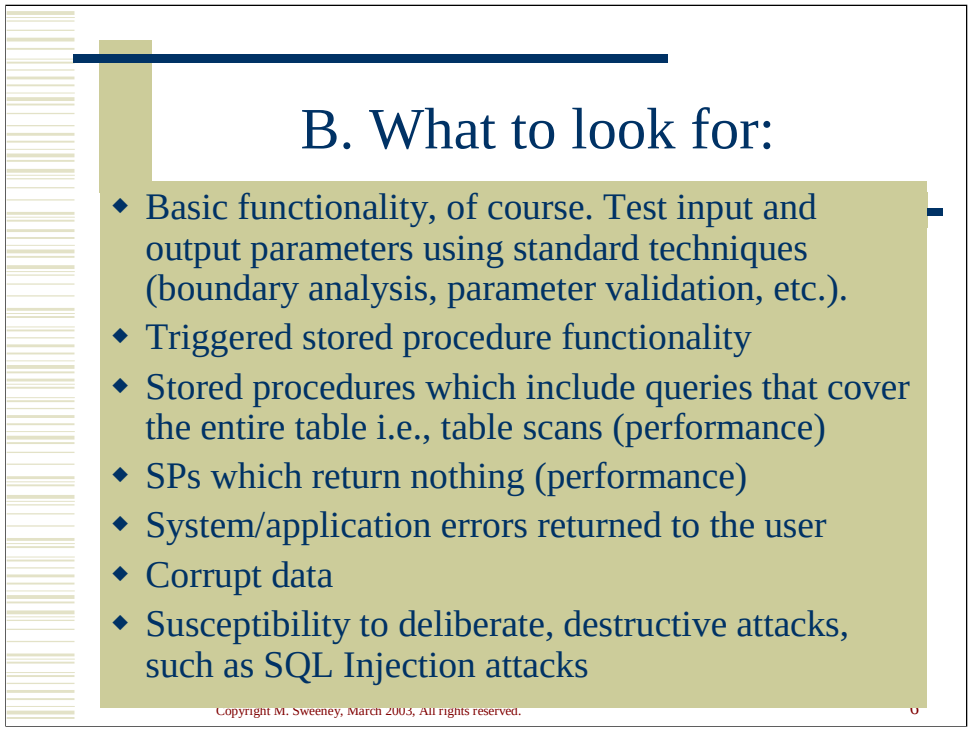
C. You can make use of your own stored procedures and use them to test the app's stored procedures and other objects as we'll see.



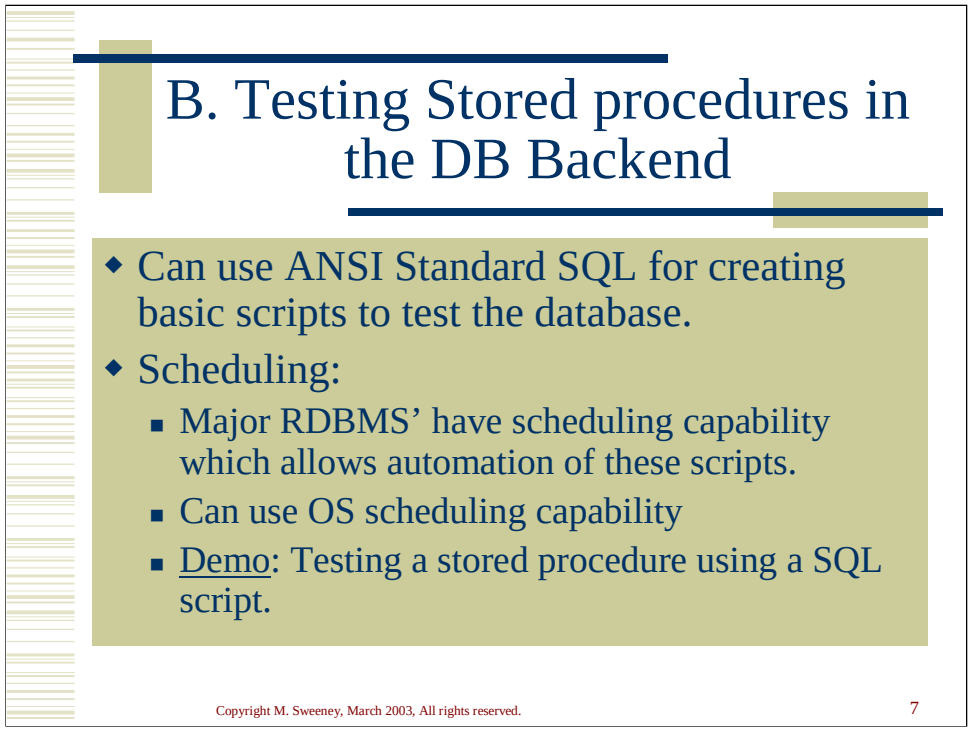

This is a short list of tools that support testing of database objects. It's by no means comprehensive. I compiled this list with my own preferred tools and those recommended anecdotally by user's groups and other colleagues. Noticeably missing from this list are the major test tool vendors. It's not that I don't recommend them, but discussions of their effectiveness are readily available. They're very expensive, but worth it if you can afford them. I've tried to focus here on tool vendors that more directly address database testing.

- DevPartner by Compuware (used to be DBPartner). It uses a nice GUI front end for exercising the database objects, including stored procedures. CompuWare has been doing this for a while and has versions for the major database products. This is advantageous because the experience of using this tool for one product is, in general, largely transferable to other products. This is key if you decide to switch to a different system, have a heterogeneous database environment, or employ people with various RDBMS backgrounds.
- I've included Visual Studio .Net, even though it's a development environment, as a tool here because it happens to be better at testing and debugging stored procedures than what was available in earlier versions. We've had the Visual Database Tools as add-ins to Visual Studio for quite a while and they didn't get nearly enough play in the testing arena as they should have. Now, the Visual database tool capabilities are upgraded and included in the platform itself. The stored procedure testing is really limited only to testing SQL Server stored procedures however. If you're using SQL Server, the Transact-SQL debugger has finally been improved enough to actually be useful!
- DBValidator by ScandiaSoft. This is a new tool by a fledgling company that's worth taking a look at. It's testing the database itself for the existence of database objects, including stored procedures. I've been on projects where transferring and transforming information from one database to another was the critical task. This tool does thing I formerly had to write myself! It belongs in any good testing toolset.
- SQL Navigator is a fairly popular tool, according to user's groups, for white box testing specific to an Oracle backend.
- If you're not aware of NUnit and JUnit, you're missing the boat! These open source tools are absolutely required for testing. They require access and knowledge of the source, so they're white box testing tools. Essentially, they allow you to instrument your code for testing. And, yes, since they're open source they're free!

If you have a favorite tool that's not on this list, please e-mail me at marys@extendtraining.com



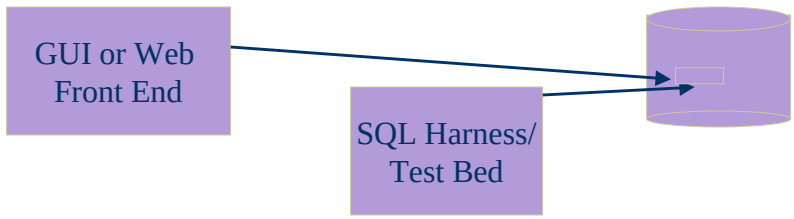

This list should give you an idea of what to look for when testing stored procedures, but, of course, it's very general. I believe you've got your hands full with recommendations of *what* to test. This presentation is more directed toward *how* to do it! Your specific system will turn out many things specific to your business requirements.




It's possible to use scripts to submit to the RDBMS that will test a variety of inputs to the critical backend stored procedures. For example, a stored procedure to insert data into the database can be tested by using a script that calls that procedure. Batch files or a table of test data can be used as the automated input to this routine. Similar scripts can be used to form a test bed for the critical database procedures. Once the test is completed and determined to be useful, it can be scheduled to run again at a future time as a batch. Major RDBMS' have scheduling capability also, in SQL Server this is called a job.

# B. Back End Testing of Stored procedures (cont)

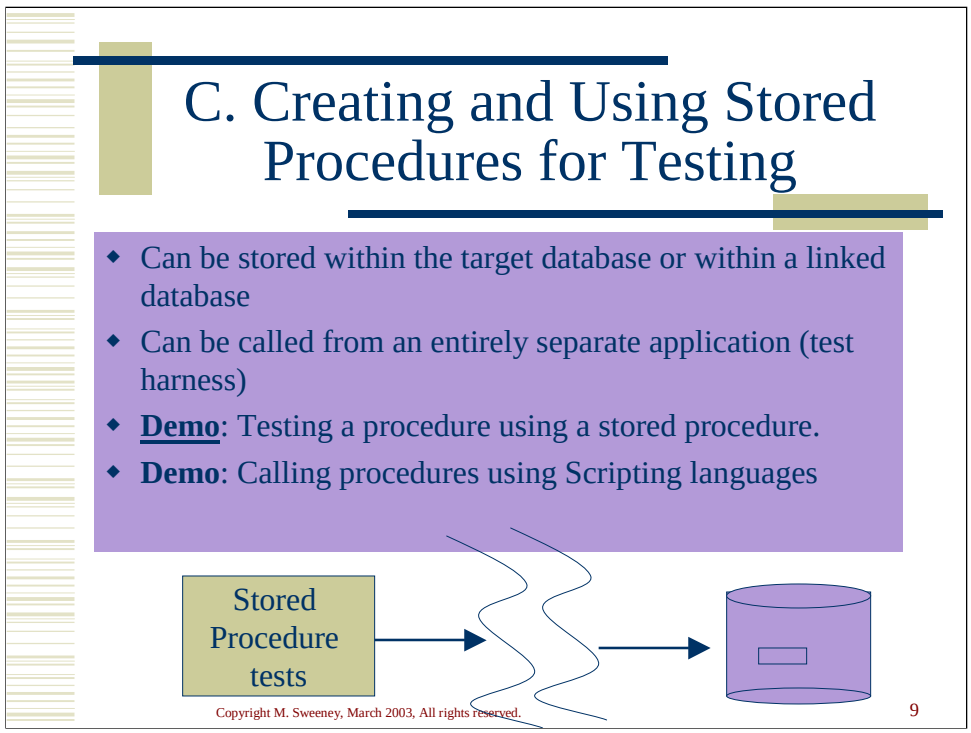
- Set up a test harness/ test bed which bypasses the front End







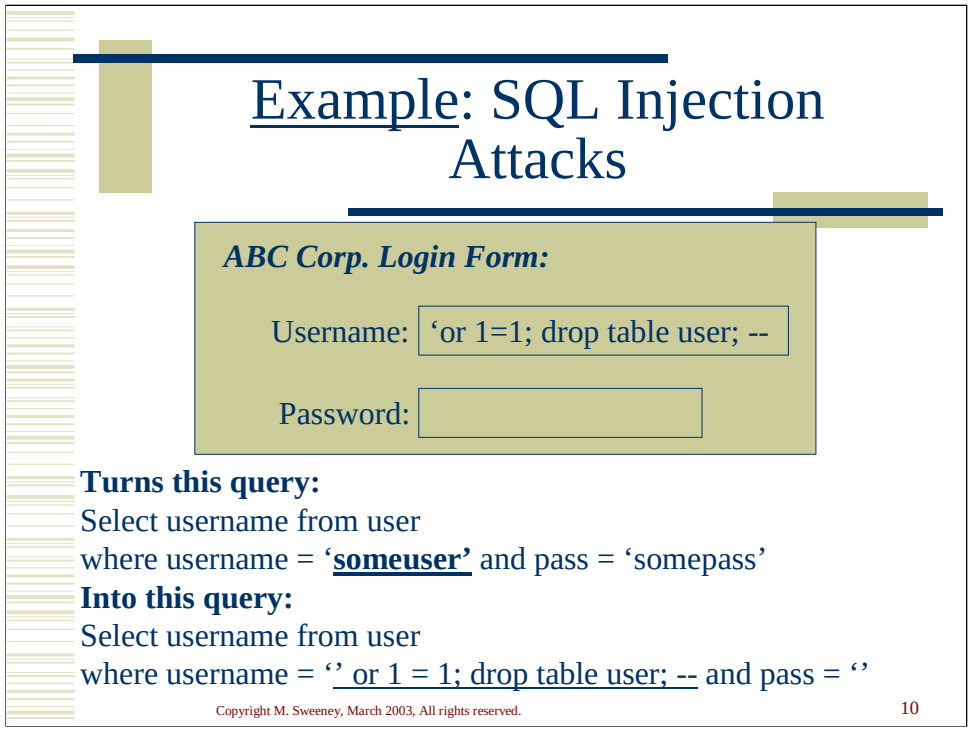
Setting up a set of routines using scripts and other stored procedures, should be done to create a test bed or a more formal harness to test basic functionality. This can and should be used in conjunction with other tools, such as NUnit/JUnit, or the higher end tools, to create a full featured test environment.




The illustration is meant to convey that the distance between the test scripts the backend is immaterial. The test stored procedures can be called from a separate application entirely. Although the test procedures will be stored within the target database, or possibly from a linked server.

The slides previous to this did not mention using stored procedures themselves to test database objects including other stored procedures and views, etc. Following along with our earlier example for the uspCustInsert stored procedure (to insert data into the database) another stored procedure to call that routine could easily be created rather than a separate script. This test procedure, created within the RDBMS SQL Code could use a loop to continuously read from a table of information and call the stored procedure using each row from the test data table submitted to the procedure as input.

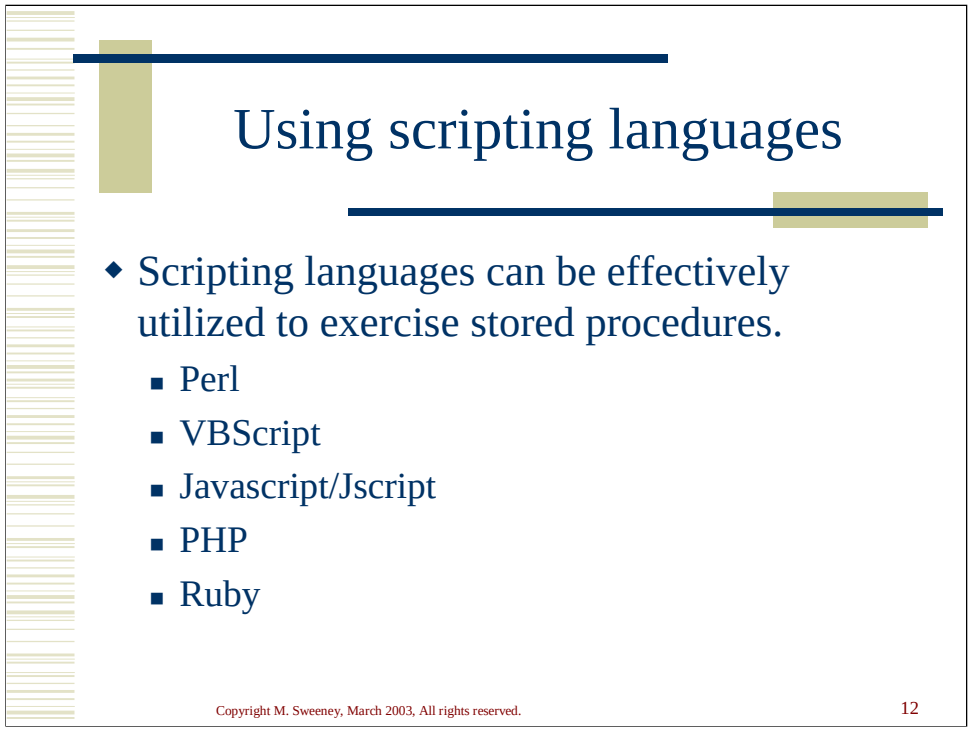
Note: In general, this recommendation should work for any RDBMS that uses ANSI-STD SQL.




SQL injection attacks can be particularly insidious. They can range from those trying to get passwords into secure systems for personal gain to those that are intentionally malicious. An excellent article about these attacks can be found at DevArticles.com http://www.devarticles.com/art/1/138. Of course the referenced article talks about what they are and how to prevent them which is important information for us, of course, but our job is to write tests to ensure that these attacks have been properly prevented. In other words we write tests to simulate a SQL-Injection attack and determine how the application handles them. It should go without saying that this needs to be performed on a test database!

# Set Up

```
Create procedure uspValidateUser
   @userName varchar(50),
   @userPass varchar(20)
as
select * from users
where userName = @userName and
   userPass=@userPass;
```




Advantages to using scripting languages for automated tests include the fact that they have a light footprint, i.e., are easy on the test system. Of course they can also directly emulate the calls being used by the application, especially if you use the same scripting language as the application. Be careful to avoid replicating application development, of course. Usually that's not an issue because the test scripts are smaller, more focused and therefore are able to isolate bugs better than using the application to do the test. For this reason, I advocate bypassing the GUI for certain important tests and using scripting languages to exercise the object.

The downside to all of this is, of course, that using scripting languages effectively requires more than a passing knowledge of how to write code. The IDE's for major programming languages providing debugging support, a help system and nice color coded screens, whereas you're on your own – for the most part -- using a scripting language. I don't believe coding in scripting languages always requires advanced programming knowledge on the part of the test engineer but, let's face it, it certainly doesn't hurt. These days, there are more technically capable testers available, as well.

# Demonstrations

- Demo: Testing a stored procedure using SQL scripts.
- Demo: Testing a stored procedure using a stored procedure.
- Demo: Test case for a SQL Injection Attack
- Demos: Calling procedures using scripting languages

# Summary

- You can and *should* use SQL and stored procedures to supplement your manual and automated tests.
- Scripting languages are useful for automated testing
- There are many ways to get started
- Now you've got some examples to start with, Go for it!

# Exceed Training

- Code examples for this presentation and the accompanying white paper are available at *www.ExceedTraining.com*
- Contact Info: marys@exceedtraining.com
- Thank you*!*





Mary can be reached at: marys@exceedtraining.com