Database Unit Testing with tSQLt

Clinton van Blommestein
Derivco

Agenda

- Who
- What
- Why
- When

About Me

- Derivco
  - Solution Architecture Team
- Senior SQL Developer
  - Casino Gameplay (Single and Multiplayer)
- OLTP performance
- ~14 years experience

For the session

- Questions welcome
- Ask me to slow down, speak louder, zoom, etc.
- Contrived examples
- Many demos

Definition of Unit Testing

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect.

Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

https://msdn.microsoft.com/en-us/library/aa292197(v=vs.71).aspx

Definition of Unit Testing

- Test a specific unit of code
- The unit is tested in isolation
- Tests are independent
- Repeatable

Why do we unit test?

- Find problems early
- Enforces business logic
  - Facilitates change
  - Documentation
- Simplify system integration
- Design

When do we add unit tests

- Before development
  - Test-driven development (TDD)
- During development
- After deployment
  - Live issue prevention

Almost every developer has written at least one unit test in their career *

Parts to unit testing

- Assemble
  - Define the test
  - Prepare the system
- Act
  - Run the Test
- Assert
  - Record the Result

Why I like unit tests

They make me look good

“Easy” for Application Developers

Many options available and integrated into IDE

- MStest
- Boost
- NUnit

Database Unit Testing

- A bit more difficult
- But not impossible

Database Unit Test Options

- SQLUnit
- dbUnit
- Visual Studio (2010 +)
- tSQLt

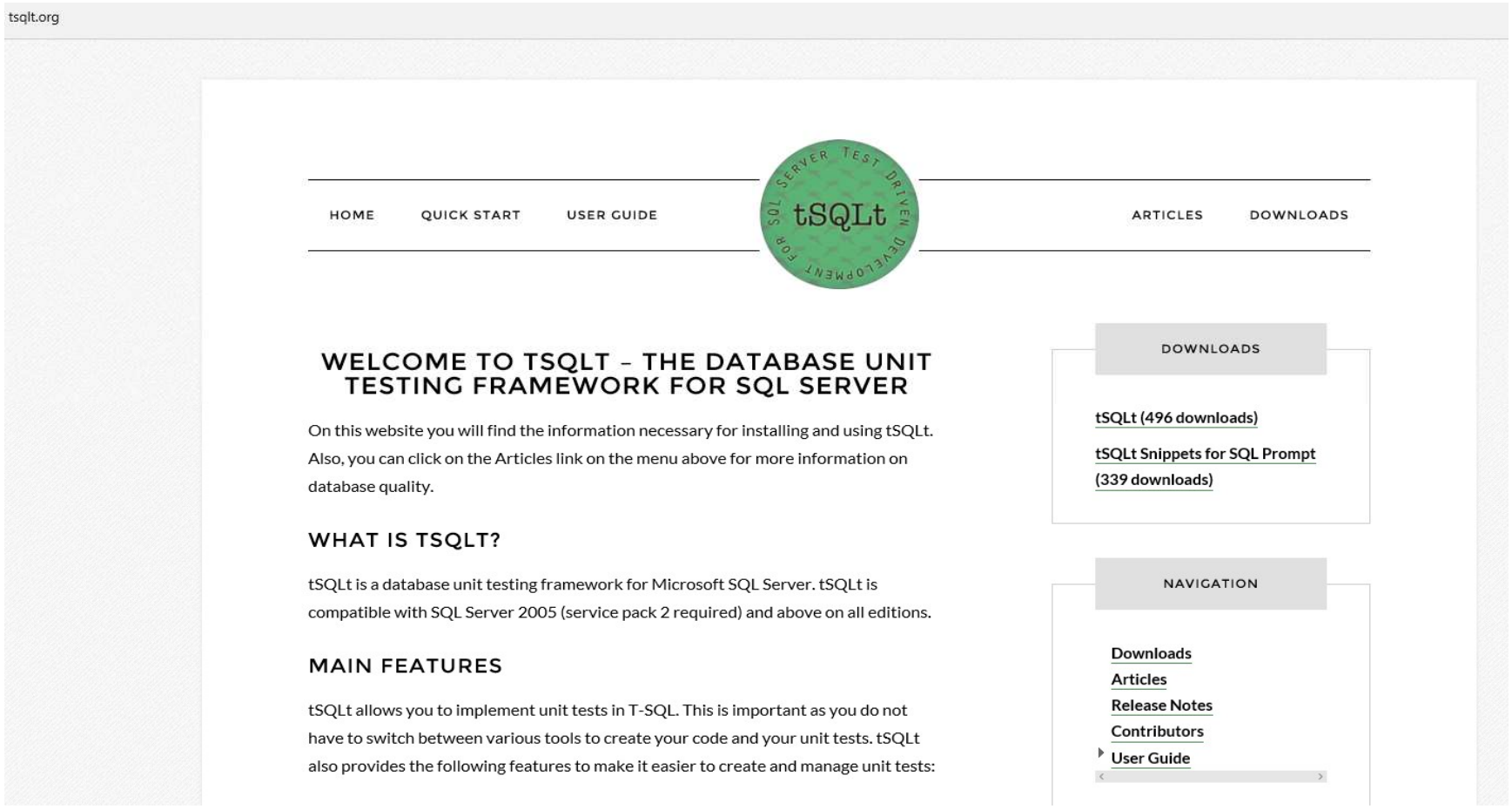
tSQLt

- Open source
- Mocking of objects (using voodoo)
- Runs at database level
  Start testing on Monday
- Includes a framework
  Red Gate SQL Test
- Has some limitations

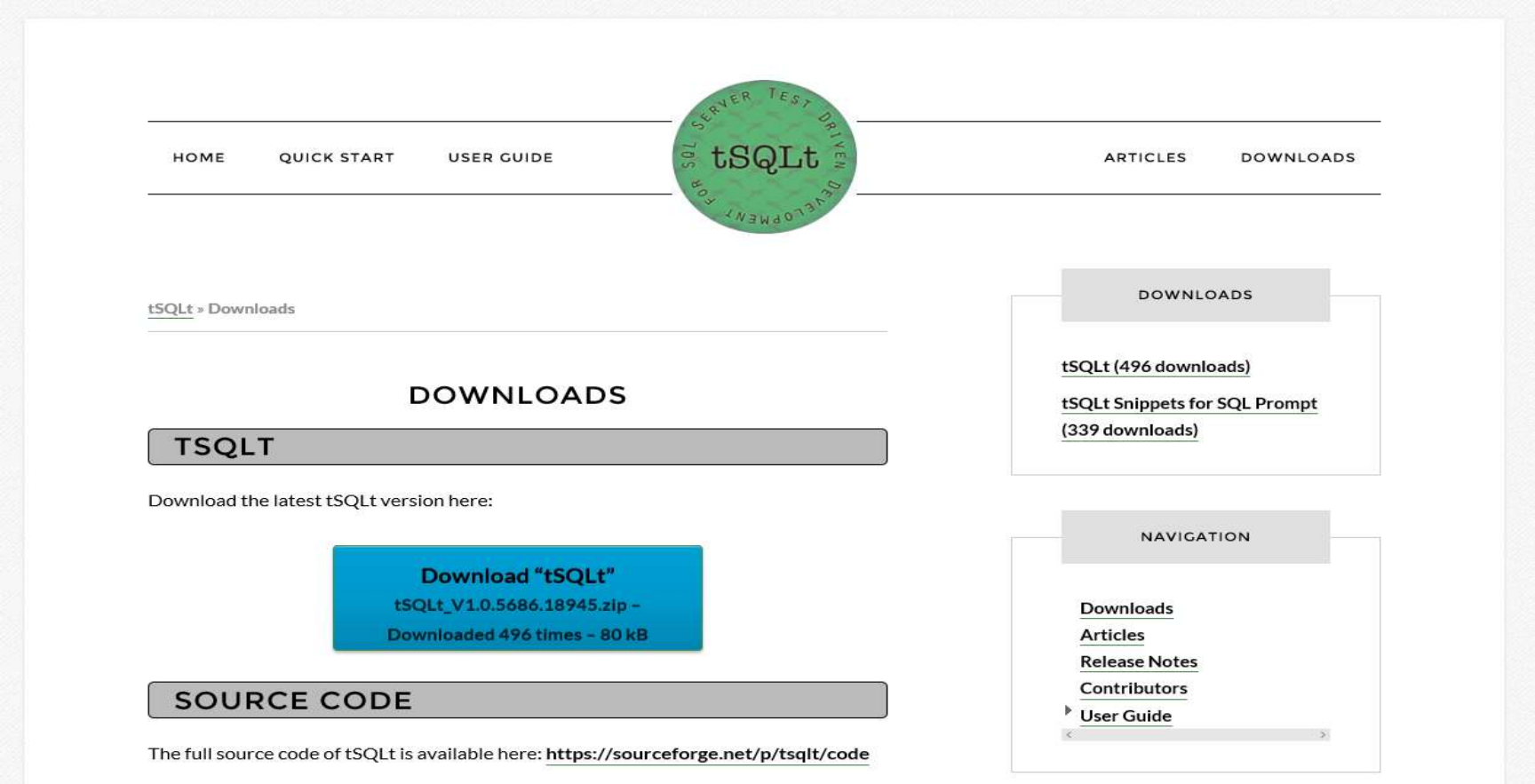
How do we use it?

- TeamCity
- Full database build, from source
- Exclusively used for tests

tSQLt


tsqlt.org
HOME
QUICK START
USER GUIDE
tSQLt
ARTICLES
DOWNLOADS
WELCOME TO TSQLT – THE DATABASE UNIT TESTING FRAMEWORK FOR SQL SERVER
On this website you will find the information necessary for installing and using tSQLt. Also, you can click on the Articles link on the menu above for more information on database quality.
WHAT IS TSQLT?
tSQLt is a database unit testing framework for Microsoft SQL Server. tSQLt is compatible with SQL Server 2005 (service pack 2 required) and above on all editions.
MAIN FEATURES
tSQLt allows you to implement unit tests in T-SQL. This is important as you do not have to switch between various tools to create your code and your unit tests. tSQLt also provides the following features to make it easier to create and manage unit tests:
DOWNLOADS
tSQLt (496 downloads)
tSQLt Snippets for SQL Prompt (339 downloads)
NAVIGATION
Downloads
Articles
Release Notes
Contributors
User Guide

tSQLt


HOME
QUICK START
USER GUIDE
tSQLt
ARTICLES
DOWNLOADS
tSQLt » Downloads
DOWNLOADS
TSQLT
Download the latest tSQLt version here:
Download "tSQLt"
tSQLt_V1.0.5686.18945.zip -
Downloaded 496 times - 80 kB
SOURCE CODE
The full source code of tSQLt is available here: https://sourceforge.net/p/tsqlt/code
DOWNLOADS
tSQLt (496 downloads)
tSQLt Snippets for SQL Prompt (339 downloads)
NAVIGATION
Downloads
Articles
Release Notes
Contributors
User Guide

tSQLt – What's in the box

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
Example.sql	176,538	37,981	SQL File	27 Jul 2015 11:31	D5C10082
License.txt	9,406	3,509	Text Document	27 Jul 2015 11:31	532E4EEA
ReleaseNotes.txt	12,401	4,491	Text Document	27 Jul 2015 11:31	7DD85D7D
SetClrEnabled.sql	775	494	SQL File	27 Jul 2015 11:31	8BC07FD4
tSQLt.class.sql	159,969	35,138	SQL File	27 Jul 2015 11:31	6B74F7C2

DEMO

Installation

First test

tSQLt - Requirements

- Requires CLR (SQL 2005+)
- Trustworthy database
- tSQLt schema
- Test names must start with "test"

DEMO

Multiple tests

Asserts

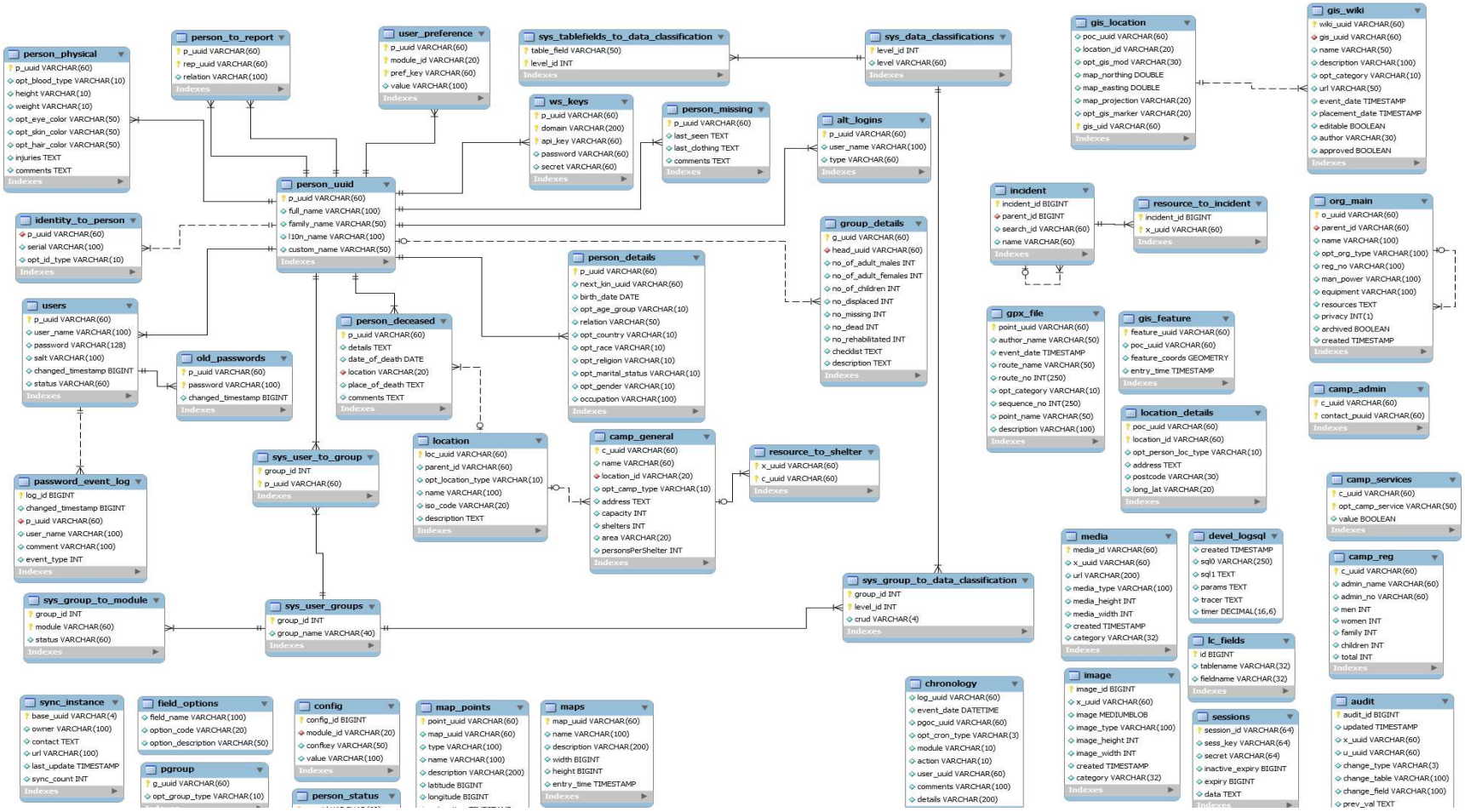
- AssertEquals
- AssertEqualsString
- AssertNotEquals
- AssertLike

- AssertObjectExists
- AssertObjectDoesNotExist

Isolating Dependencies

- Tables
- Stored Procedures
- Functions

Isolating Tables

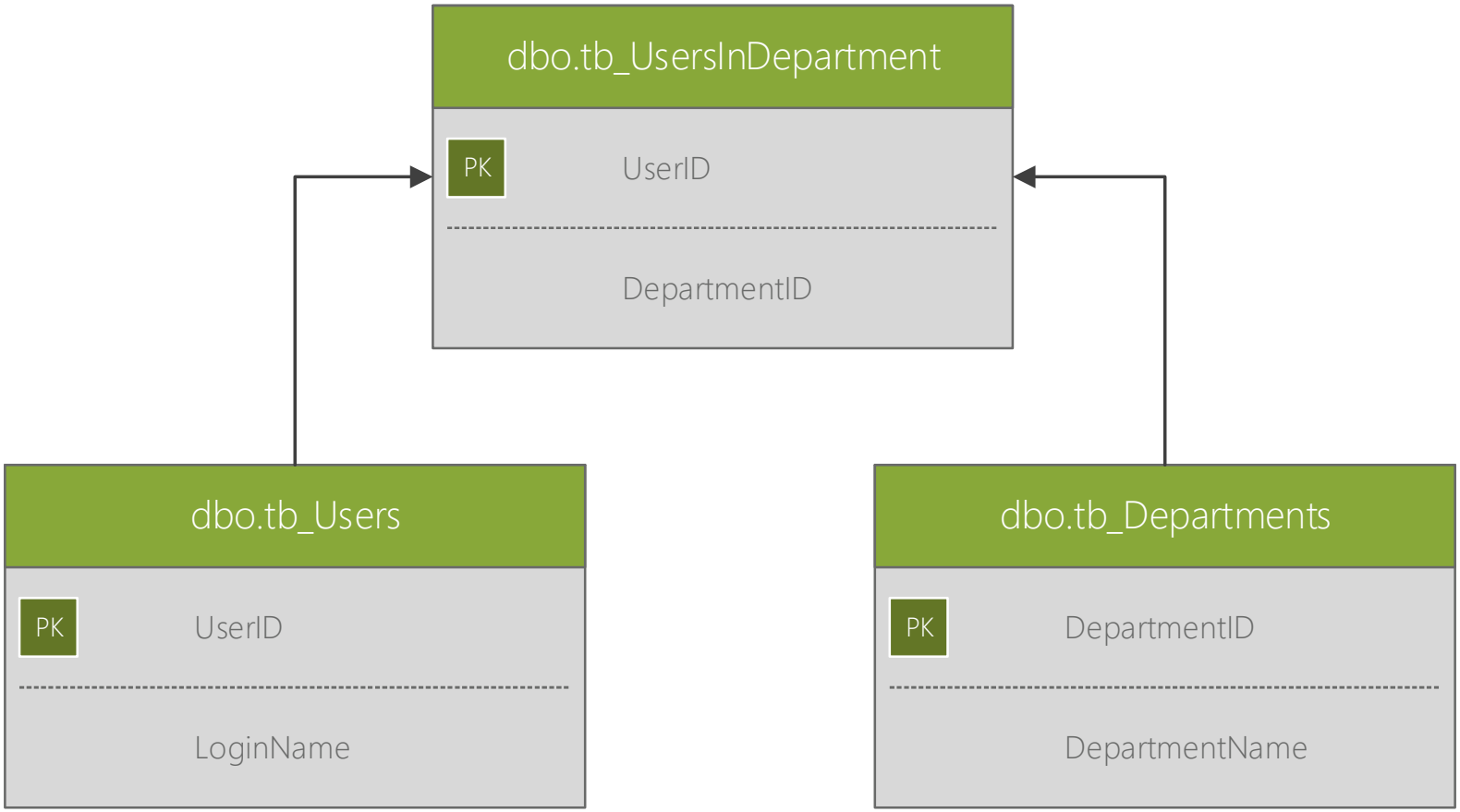


PASS SQL saturday

The Hard Stuff

- Foreign Keys
- Constraints
- Identities

Isolating Tables



Business rules for Login

- Can only log in with correct password
- After incorrect password user cannot log in for 5 minutes
- Some form of auditing

Isolating Tables

tSQLt.FakeTable

@TableName
@SchemaName
@Identity
@ComputedColumns
@Defaults

DEMO

Faking a table

More Asserts

FakeTable Limitations

- Objects cannot be schema bound, or in-memory
- Table cannot be published
- Tests must be updated when table schema changes
- Not all datatypes fully supported

Asserts

- AssertEmptyTable
- AssertEqualsTable

- AssertResultSetsHaveSameMetaData

WARNINGS

- Not for LIVE or SHARED environments
- Be careful of FakeTable tests

WARNING

Help, my tables are gone

They are still there

```
SELECT
      OriginalName,
      SCHEMA_NAME(schema_id) + '.' + name AS [Name of Renamed Table],
      create_date
FROM tSQLt.Private_RenamedObjectLog
 JOIN sys.objects
  ON objectid = object_id;
```

	OriginalName	Name of Renamed Table	create_date
1	[tb_Users]	dbo.tSQLt_tempobject_a4028c0519a04ba196904123392...	2015-08-30 16:39:05.700
2	[tb_Departments]	dbo.tSQLt_tempobject_0d5809840b0f4fb1b50dd4633a4c...	2015-08-30 16:39:05.697
3	[tb_UsersInDepartment]	dbo.tSQLt_tempobject_0d515b996f8d40bf8fd9dc5a515ba...	2015-08-30 16:39:05.703

Transaction Control

- tSQLt creates its own transaction
- Your objects needs to be transaction aware
- Careful running tSQLt commands outside of controlled test

DEMO

Transaction Control

Isolating Stored Procedures

- Stored procs
  - calling stored procs
    - calling stored procs

Isolating Stored Procedures

tSQLt.SpyProcedure

@ProcedureName
@CommandToExecute

DEMO

Spy Procedure

SpyProcedure Limitations

- RETURN code always 0
- No # (temp) stored procedures
- GETDATE() and linked servers
- Tests can look "messy"
- Encourages small, specific procedures

Isolating Functions

- Similar but different to SpyProcedure
- Stored Procedure
  calling function

Isolating Functions

tSQLt.FakeFunction

@FunctionName
@FakeFunctionName

Fake Function Limitations

- A real function needs to be created
- Original and fake must be of same type
  - Scaler & Scaler
  - Table Valued & Table Valued
  - Result set meta data

What to look out for

- Meaningful tests
- Long running tests

Thanks

Questions?

clinton.vb@derivco.com