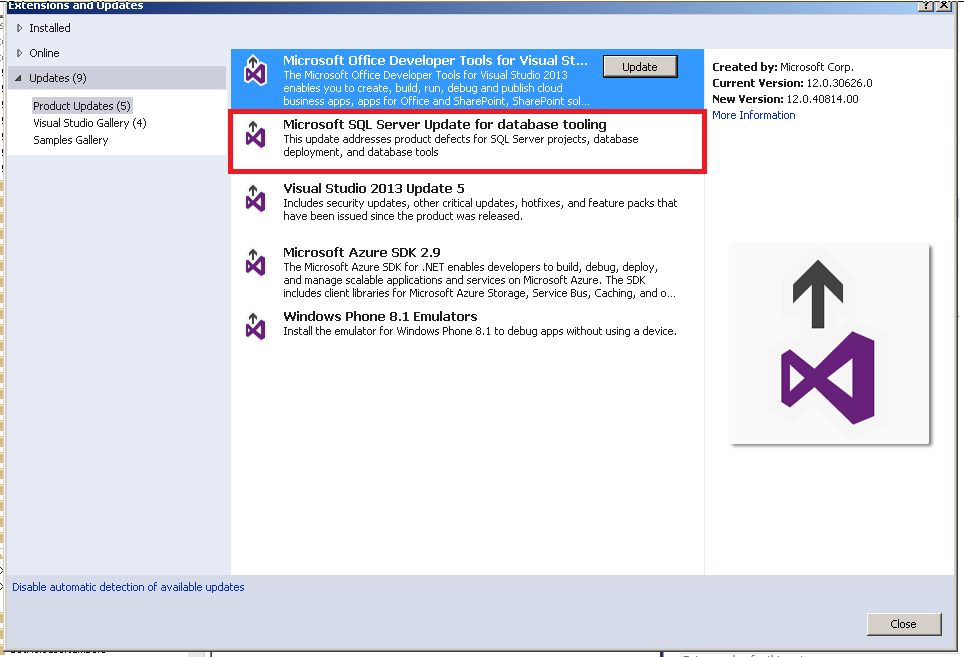
1. Setup Environment

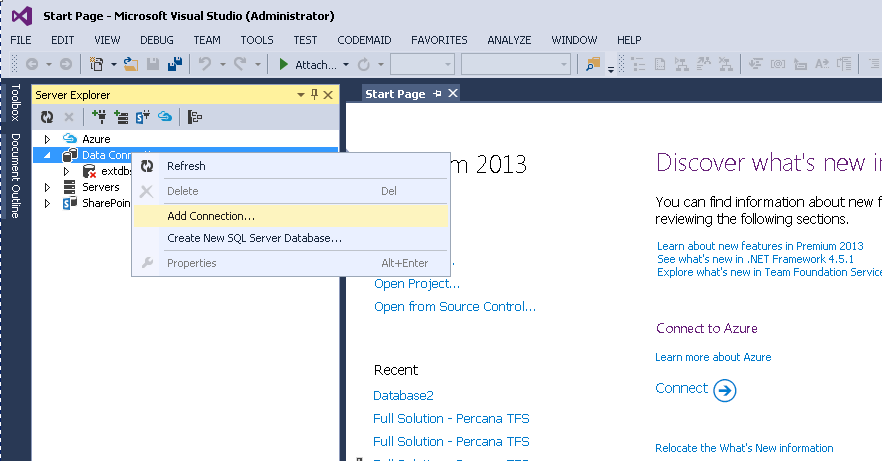
For Visual Studio 2013, the SQL unit test component is ready and just open [Tools] -> [Extensions and Updates] to install the SQL update if the update exists.

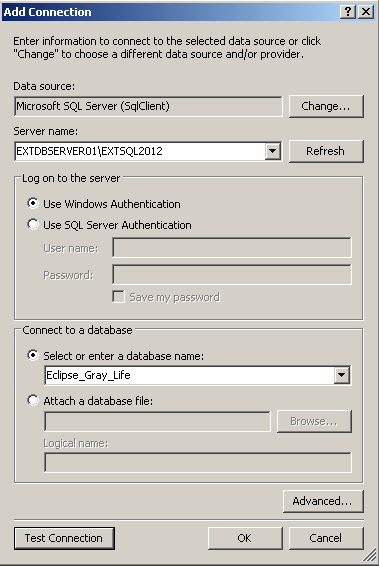


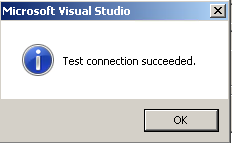
The update also should be checked if ready for Visual Studio 2012.

1. Connect Database

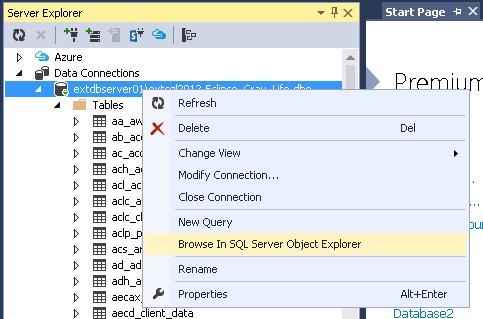
Add SQL server connection.

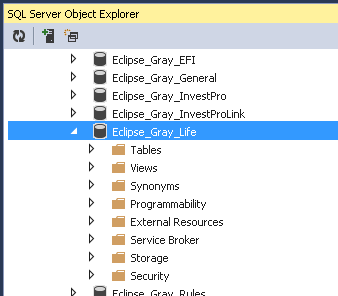




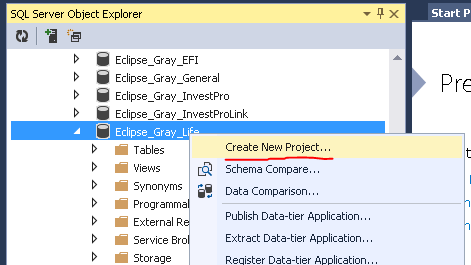


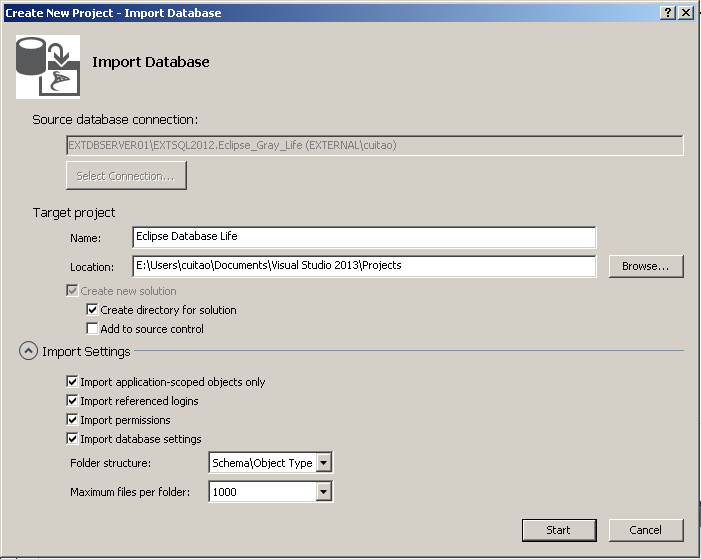
1. Create Database Project
2. Open SQL db object





1. Create a new database project for selected db.

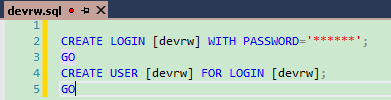
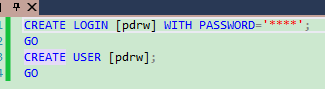




1. Resolve Database Project Issues
   1. Security

 Or 

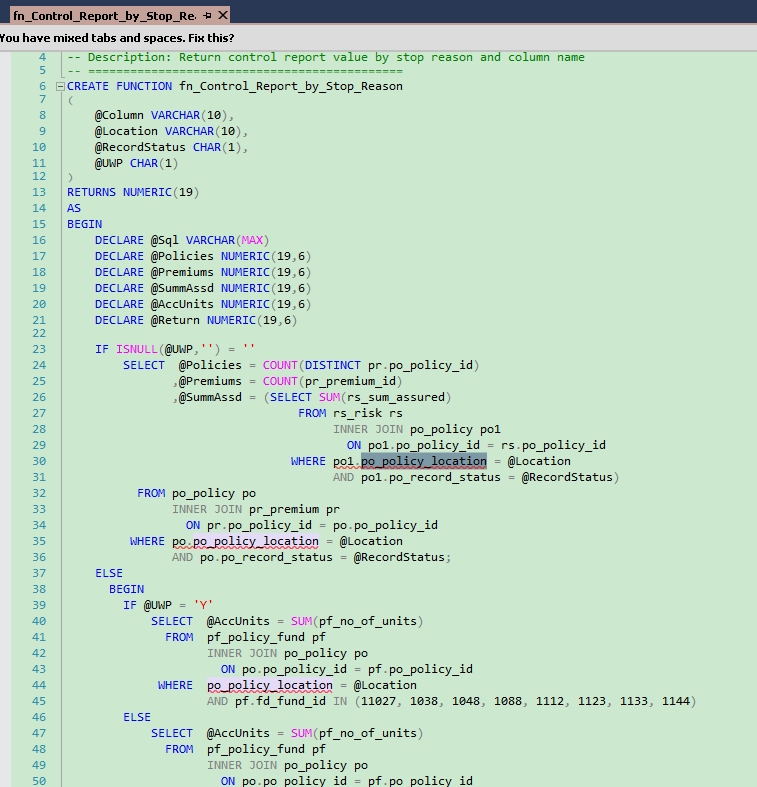
In this sample, the **USER** devrw need a **LOGIN** name. So, **LOGIN** devrw should be created before create user statement.

Or 

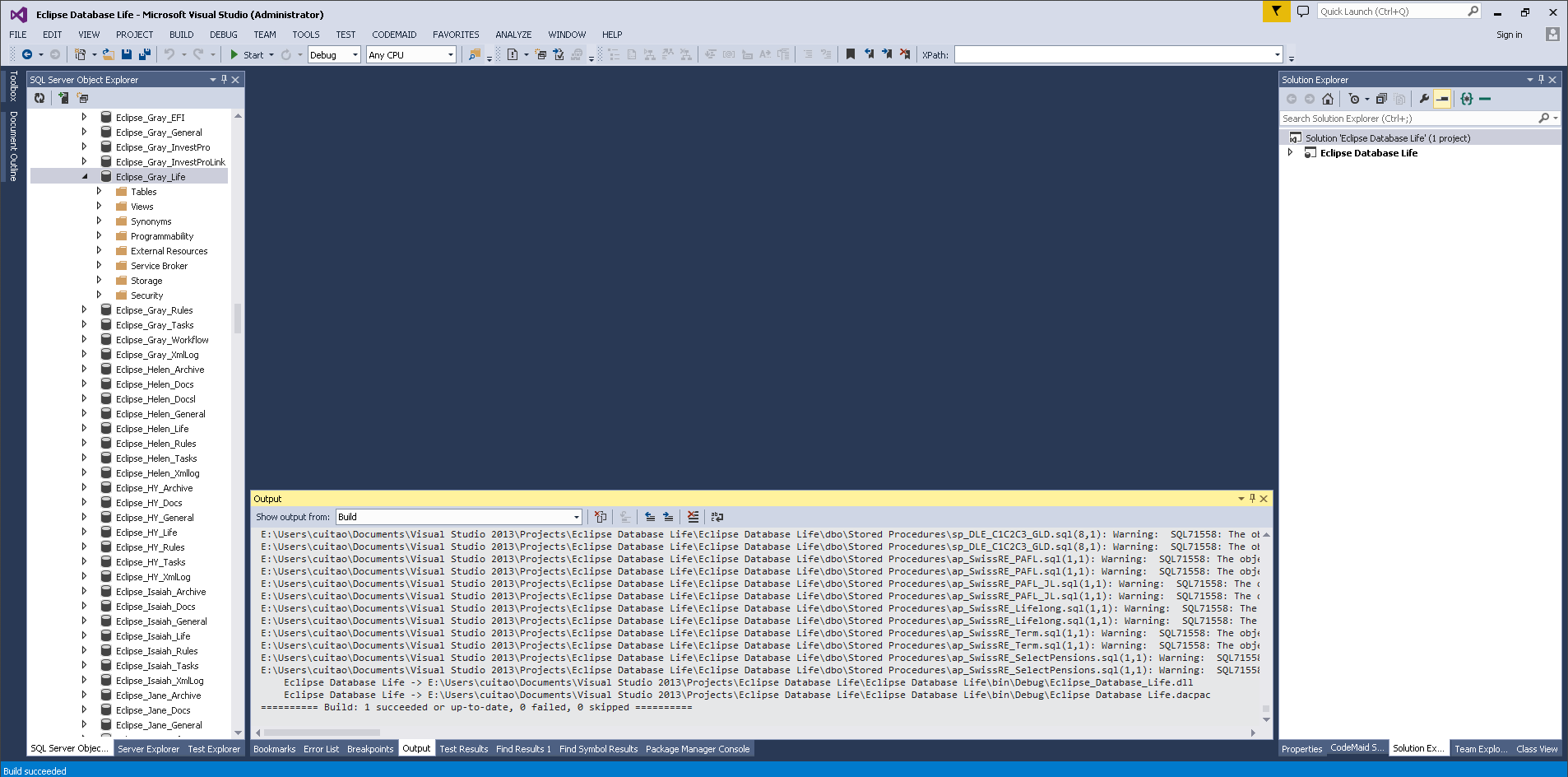
* 1. dbo

For Functions, Store Procedures, and View, just exclude files, which reference non-existing columns, from project.

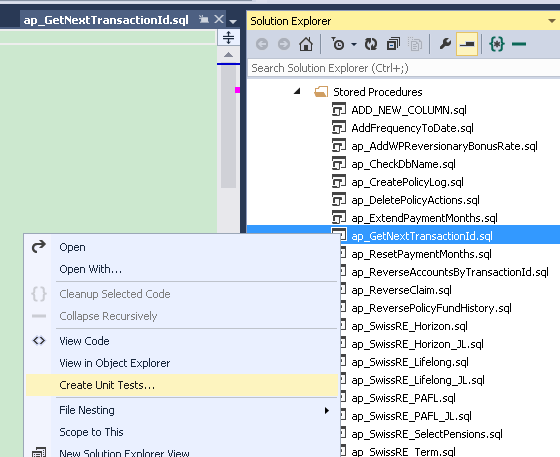
For example, the function “fn\_Control\_Report\_by\_Stop\_Reason.sql” reference column “po\_policy\_location” which doesn’t exist in life db. Just remove that kind of files from project for successful building.

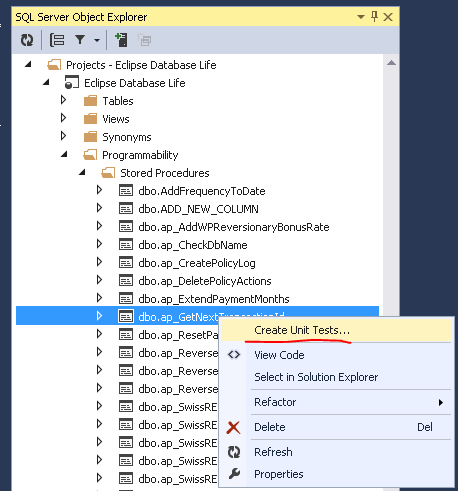


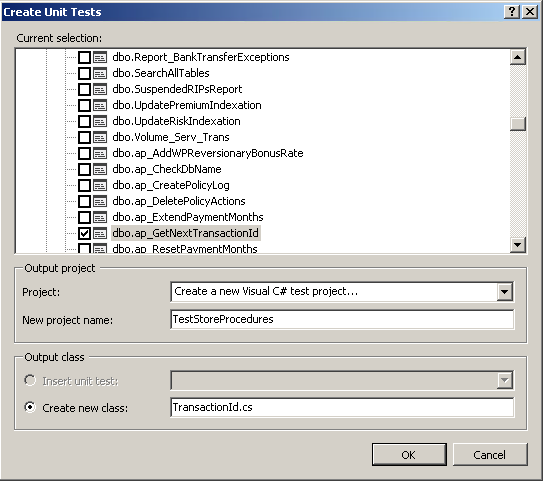
1. Build Project and Create Unit Test
   1. Build Project



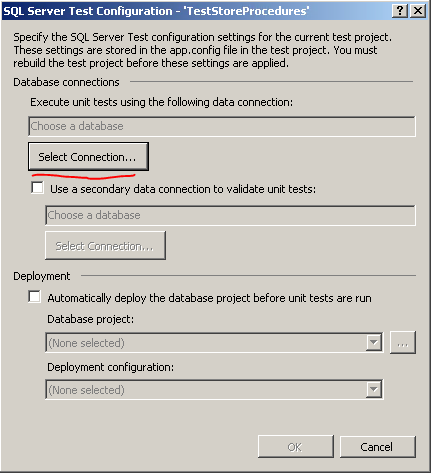
* 1. Create Unit Test

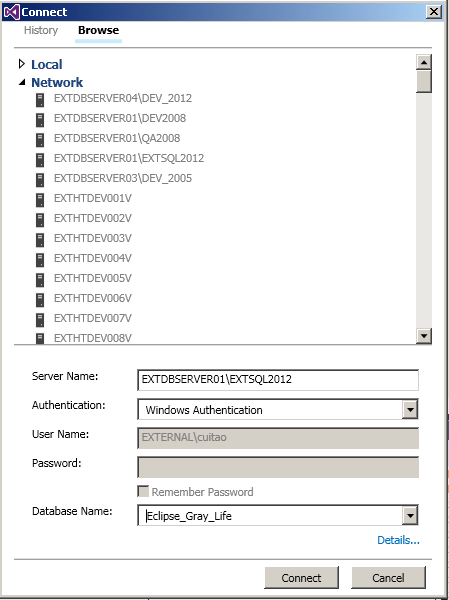




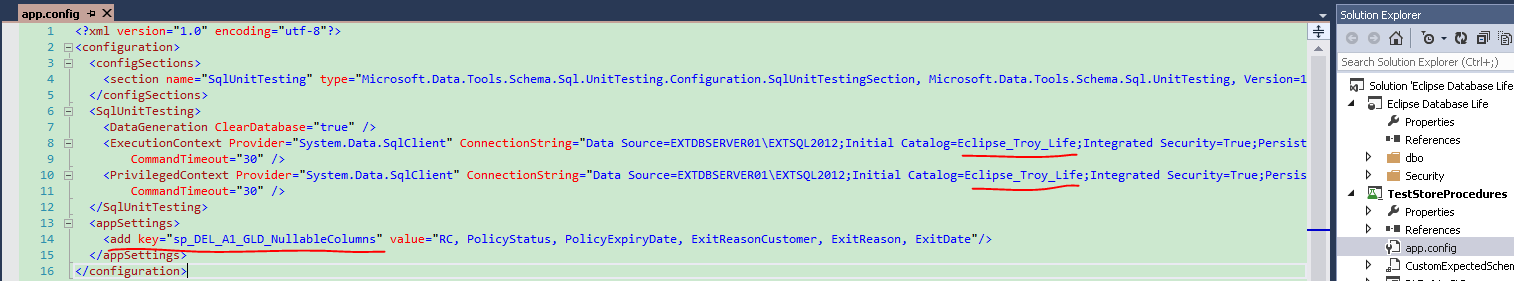


Create a SQL server connection for unit test project.





* 1. Config SQL Connection and Add Custom AppSettings

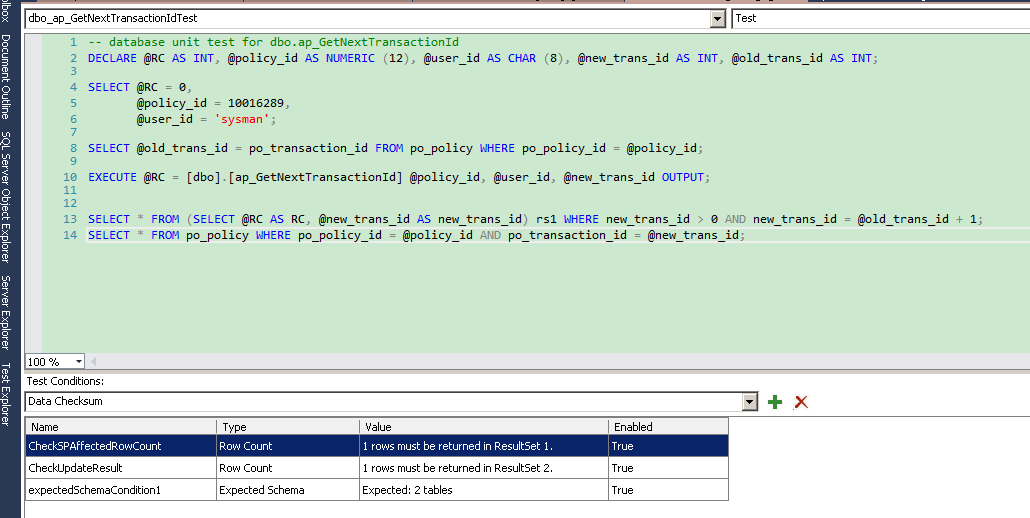


1. Edit ExecutionContext.ConnectionString to execute SQL script on target db.
2. Add appSettings if necessary.
3. Edit Test Action
   * 1. SQL Script Editor

Double click SQL unit test file to open design window. Edit SQL script in top editor area to return what wanted.

In this sample, the test action return two result set.

1. Result 1: 
2. Result 2: 



* + 1. Test Condition Editor.

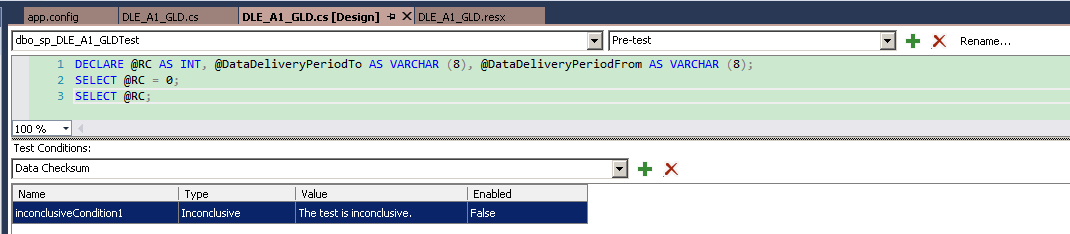
Please refer <https://msdn.microsoft.com/en-us/library/jj851223%28v=vs.103%29.aspx>

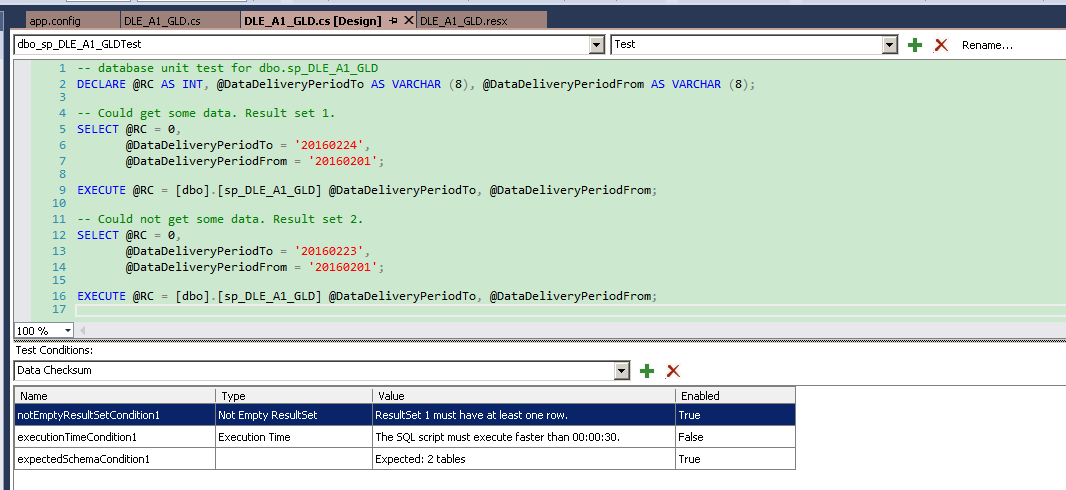
|  |  |
| --- | --- |
| **Inconclusive** | Always produces a test with a result of Inconclusive. This is the default condition added to every test. This test condition is included to indicate that test verification has not been implemented. Delete this test condition from your test after you have added other test conditions. |

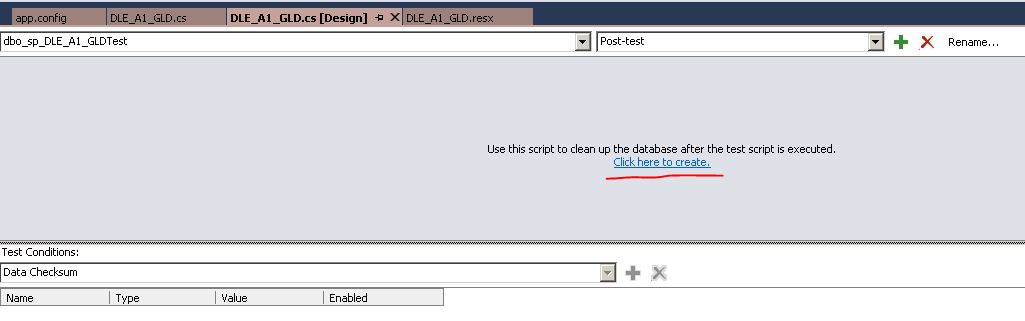
* + 1. Config Test Action

SQL unit test have three types of test action- Pretest Action, Test Action and Posttest Action.

Script of them could be edited on UI.





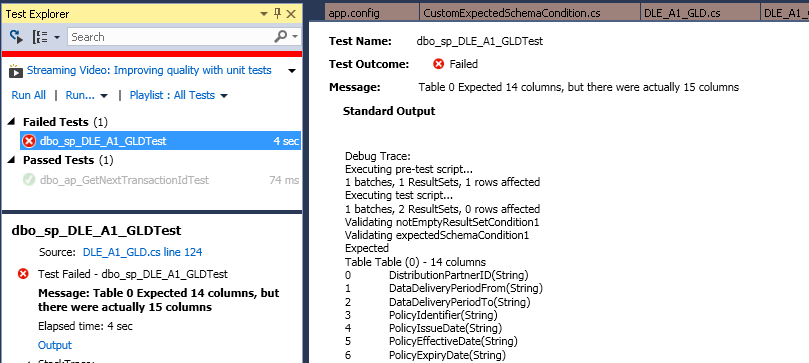


1. Create Custom Test Condition

Test condition could be inherited and add additional code to do more.

In this sample, the expected schema condition is extended to show how to write custom test condition.

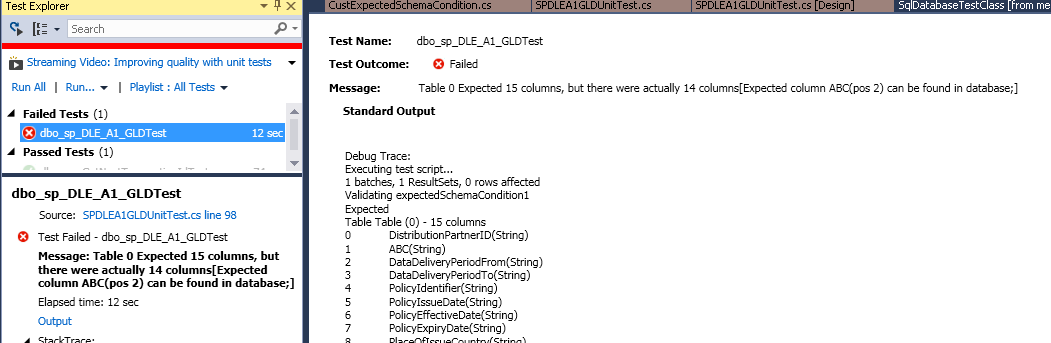
Generally, the failure message of expected schema condition doesn’t contain the column difference details between db schema and test schema.



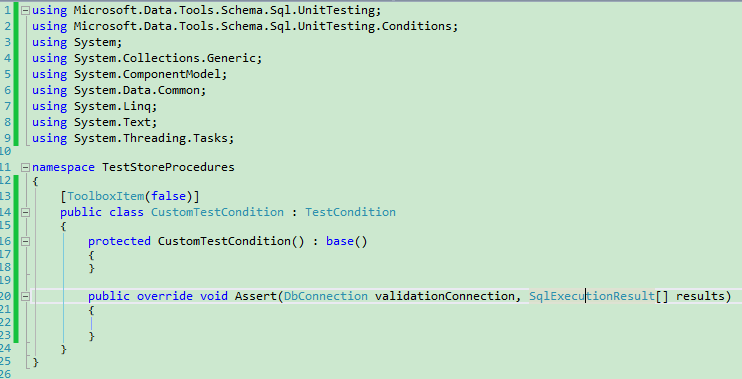
To avoid find the differences between both schema manually, a new class CustomExpectedSchemaCondition is added to compare and find all differences.



1. Override the assert method of expectSchemaCondition.
2. Call the assert method of expectSchemaCondition first, when expectSchemaCondition asserting failed, get columns from SQL execution result object and expected schema condition object.
3. Find all differences and build new failure message. Like output below, a new column ABC have been added to sp\_DLE\_A1\_GLD, but the result set from db doesn’t contain the new column. That detail is reported by custom expected schema condition.



* 1. Create General Custom Test Condition



1. Add Method to Prepare Test Data

The connection object has been initialized before call test method, so the object could be used directly to execute some SQL query to prepare data at the first of test method.



1. Add Non Test Condition Method to Verification Result Set Further

For example, verify if all rows are not null or empty. After all test conditions are passed, the result would be verified further. In this sample, method validateEmptyCell get result set and check if there are null or empty values under a column which is not allowed to be null or empty.

