

Unit Testing for Database Development

Olivier Van Steenlandt



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

17th edition SQLDay Conference

12-14 May 2025, WROCŁAW + ONLINE



Platinum sponsors



Gold sponsors



Silver sponsors



About me

Olivier Van Steenlandt



Expert @ datashift



Core Member @ dataMinds.be



Speaking / Blogging



Running / Cycling / Swimming



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Schedule

What

Why

How

Demo

Summary



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Current Situation

Environments



Development
(Acceptance)
Production

Process



Automatic:
* Change tracking
* Deployments

Tools



Database Project
Azure DevOps



DATA-CUISINE.COM
BITE SIZE DATA TALES



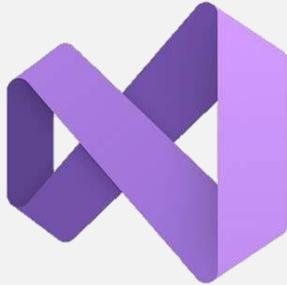
/in/oliviervansteenlandt



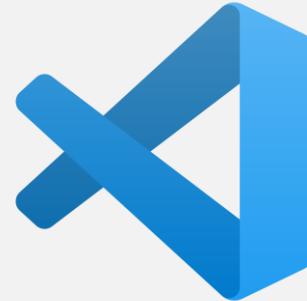
@Oli_VSteenlandt

@oliviervs.bsky.social

Database Projects



Visual Studio



Visual Studio
Code



Azure Data
Studio



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt

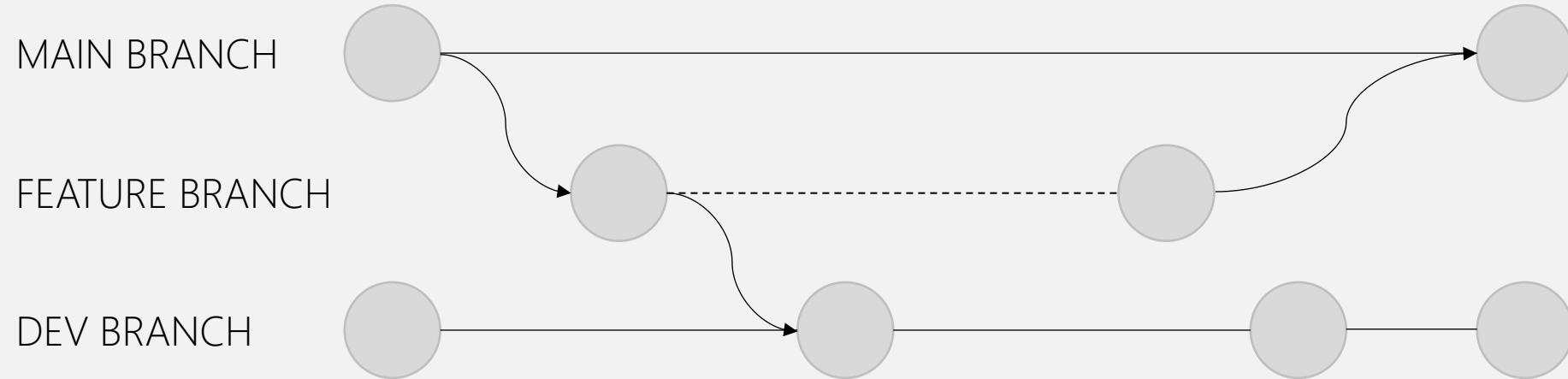


@Oli_VSteenlandt

@oliviervs.bsky.social

Branching Strategies

Feature Branching



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Future Situation

Process



Automatic
Change tracking
Deployments
Unit Testing

Tools



Database Project
Azure DevOps
Unit Testing Framework



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt

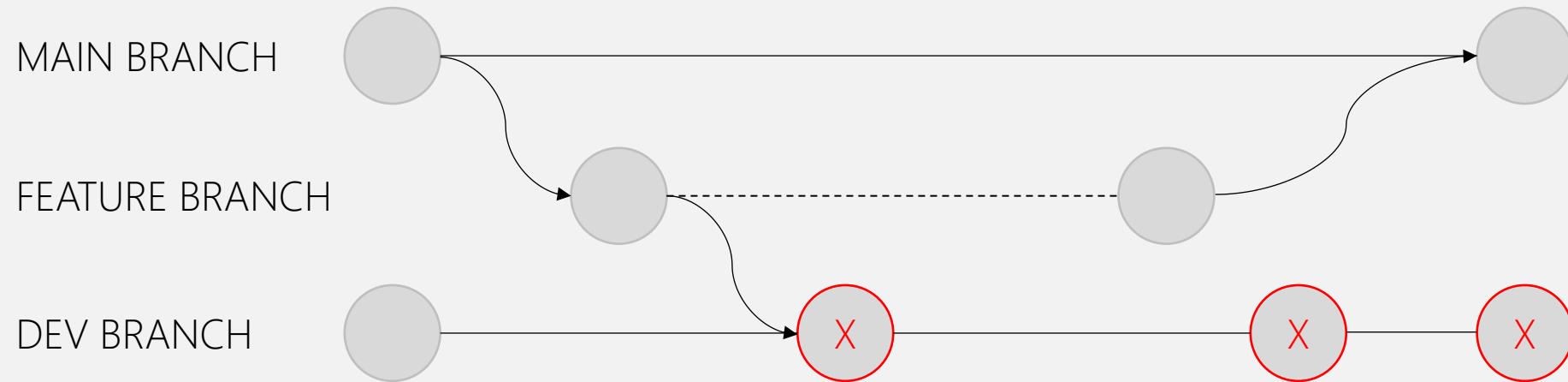


@Oli_VSteenlandt

@oliviervs.bsky.social

Branching Strategies

Feature Branching



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Concepts & Basics



DATA-CUISINE.COM
BITE SIZE DATA TALES



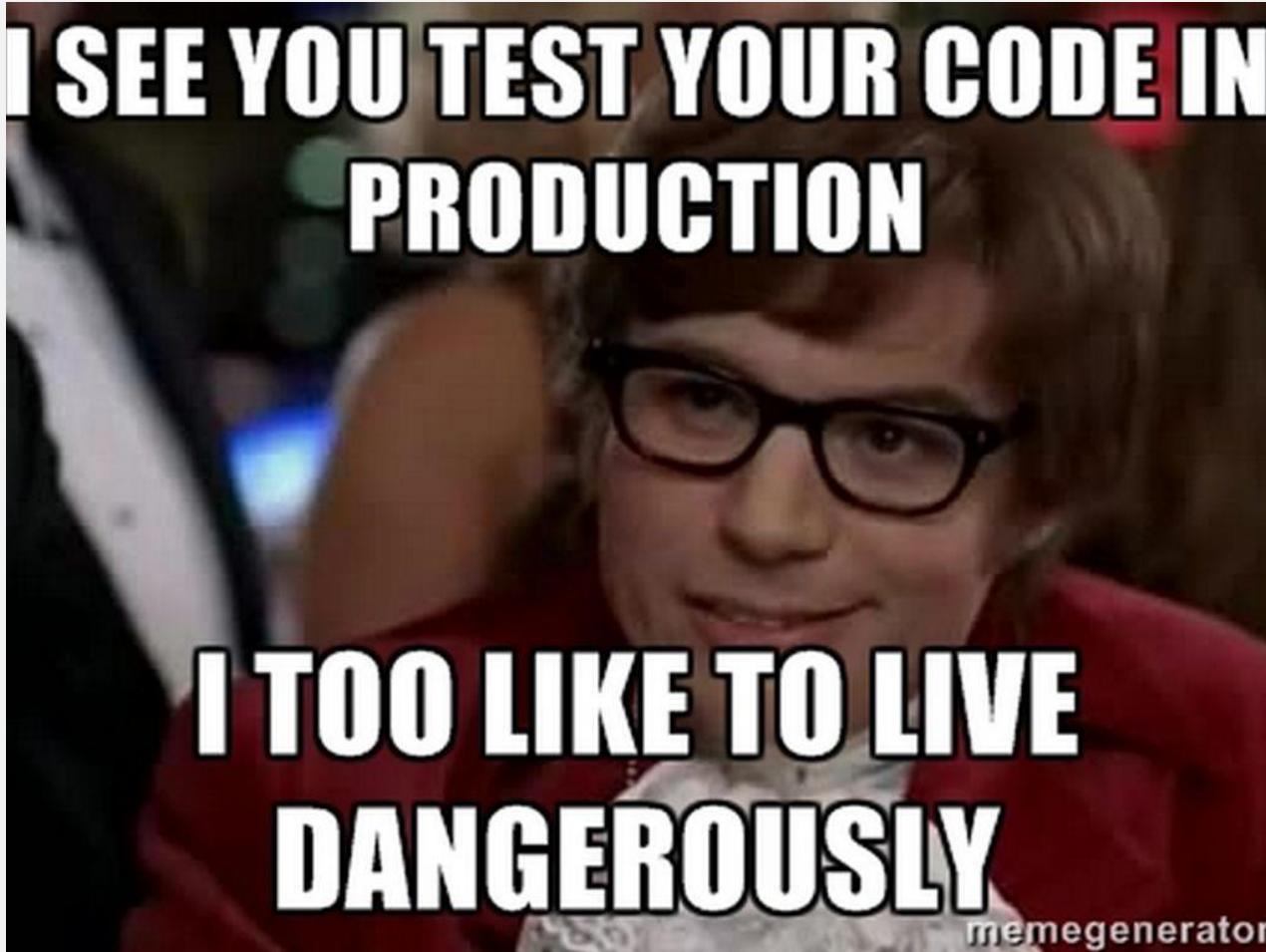
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Why?



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Why?

Identify bugs early

Improves Code Quality

Cost Reduction Long term



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

General

UI Testing

Unit Testing

Integration Testing



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

What is Unit Testing

“Unit Testing is a form of software testing by which isolated source code is tested to validate expected behavior” - *Wikipedia*



Typical Design Pattern

1. Create a Test File
2. Write a Test Class
3. Execute Test File
4. Validate the result



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Unit Testing Framework

Junit – Java

Jest – JavaScript

PyUnit – Python

Xunit - .Net

Nunit - .Net



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Unit Testing for SQL Databases



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Options for Database Unit Testing

Create your own ☺

SQL Server Unit Tests

tSQLt



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

tSQLt



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

tSQLt

<https://tsqlt.org/>

→ Sebastian Meine

→ Liz Baron



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

How to get started?

Download tSQLt

Install tSQLt

Write your first Unit Test



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt

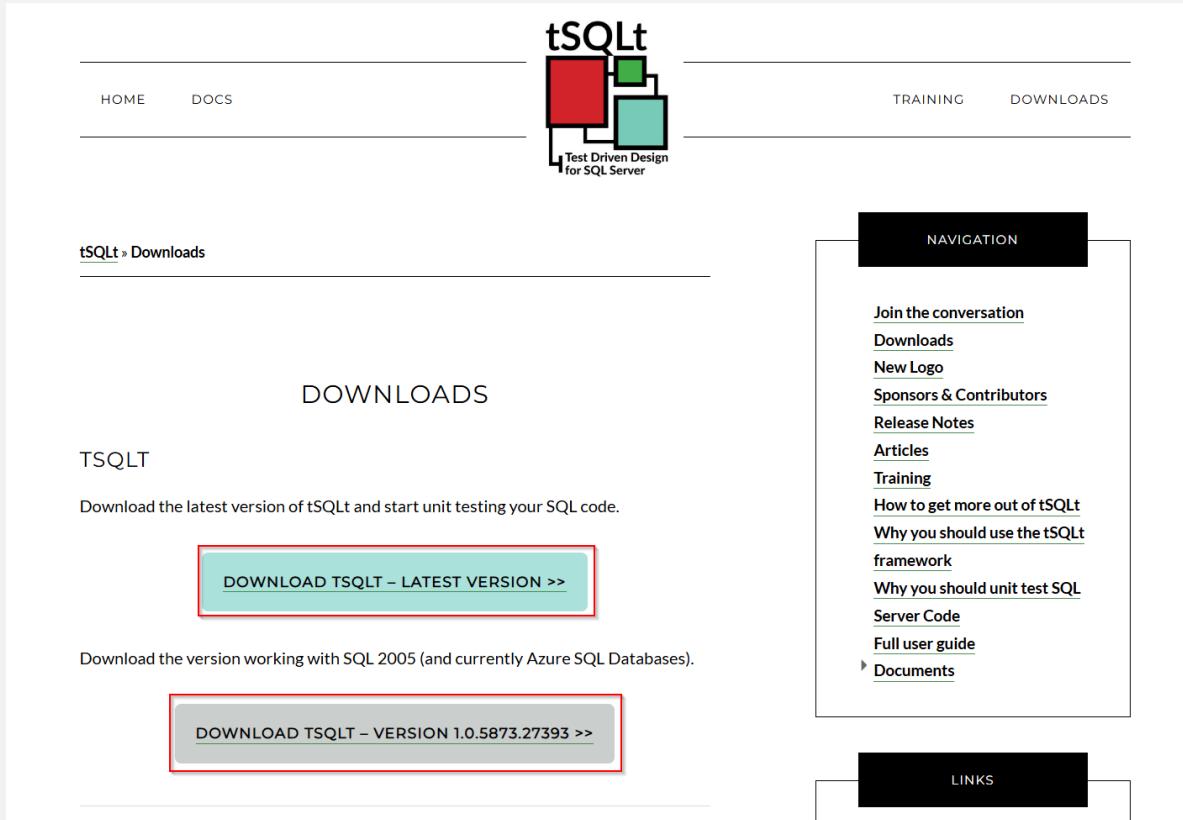


@Oli_VSteenlandt

@oliviervs.bsky.social

Download tSQLt

<https://tsqlt.org/downloads>



The screenshot shows the 'Downloads' section of the tSQLt website. At the top, there's a navigation bar with links for 'HOME', 'DOCS', 'TRAINING', and 'DOWNLOADS'. The central logo features the text 'tSQLt' above a graphic of overlapping red, green, and teal squares, with the subtitle 'Test Driven Design for SQL Server' below it. Below the navigation, a breadcrumb trail reads 'tSQLt » Downloads'. The main content area is titled 'DOWNLOADS' and contains a section for 'TSQLT'. It encourages users to 'Download the latest version of tSQLt and start unit testing your SQL code.' A prominent blue button with white text says 'DOWNLOAD TSQLT – LATEST VERSION >>'. Below this, another section offers the 'SQL 2005 (and currently Azure SQL Databases)' version, with a grey button labeled 'DOWNLOAD TSQLT – VERSION 1.0.5873.27393 >>'. To the right, a 'NAVIGATION' sidebar lists various links: 'Join the conversation', 'Downloads', 'New Logo', 'Sponsors & Contributors', 'Release Notes', 'Articles', 'Training', 'How to get more out of tSQLt', 'Why you should use the tSQLt framework', 'Why you should unit test SQL', 'Server Code', 'Full user guide', and 'Documents'. At the bottom right of the page is a 'LINKS' section.



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Available versions

tSQLt_V1.0.8083.3529

→ SQL Server

tSQLt_V1.0.5873.27393

→ Azure SQL Database



DATA-CUISINE.COM
BITE SIZE DATA TALES



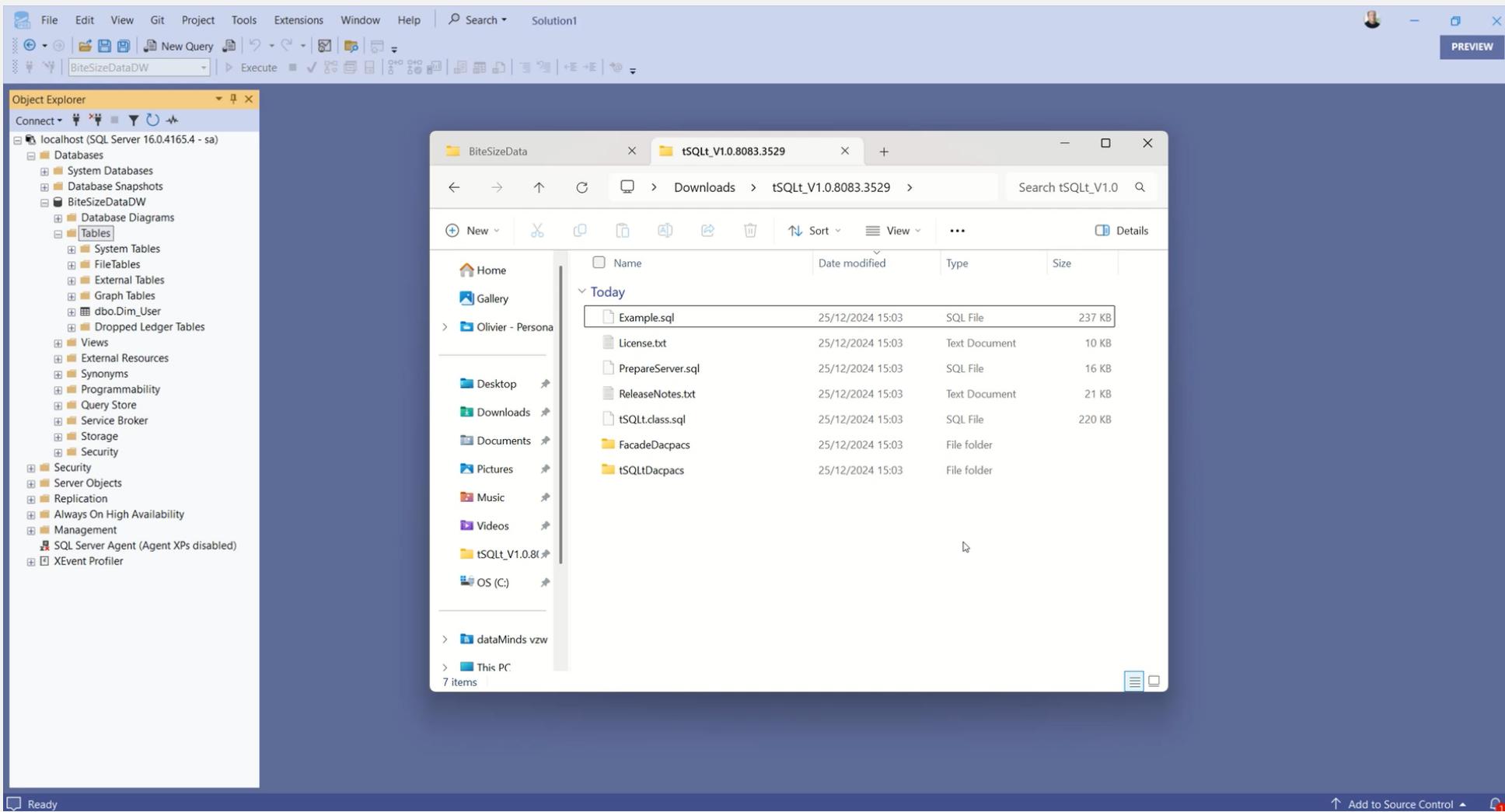
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Installing tSQLt



tSQLt Installed, now what?

1. Create a Test Class

Grouping of Unit Tests

2. Create a Unit Test

Testing specific functionality



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Create a Test Class

tSQLt.NewTestClass @ClassName

→ Stored Procedure

→ Example:

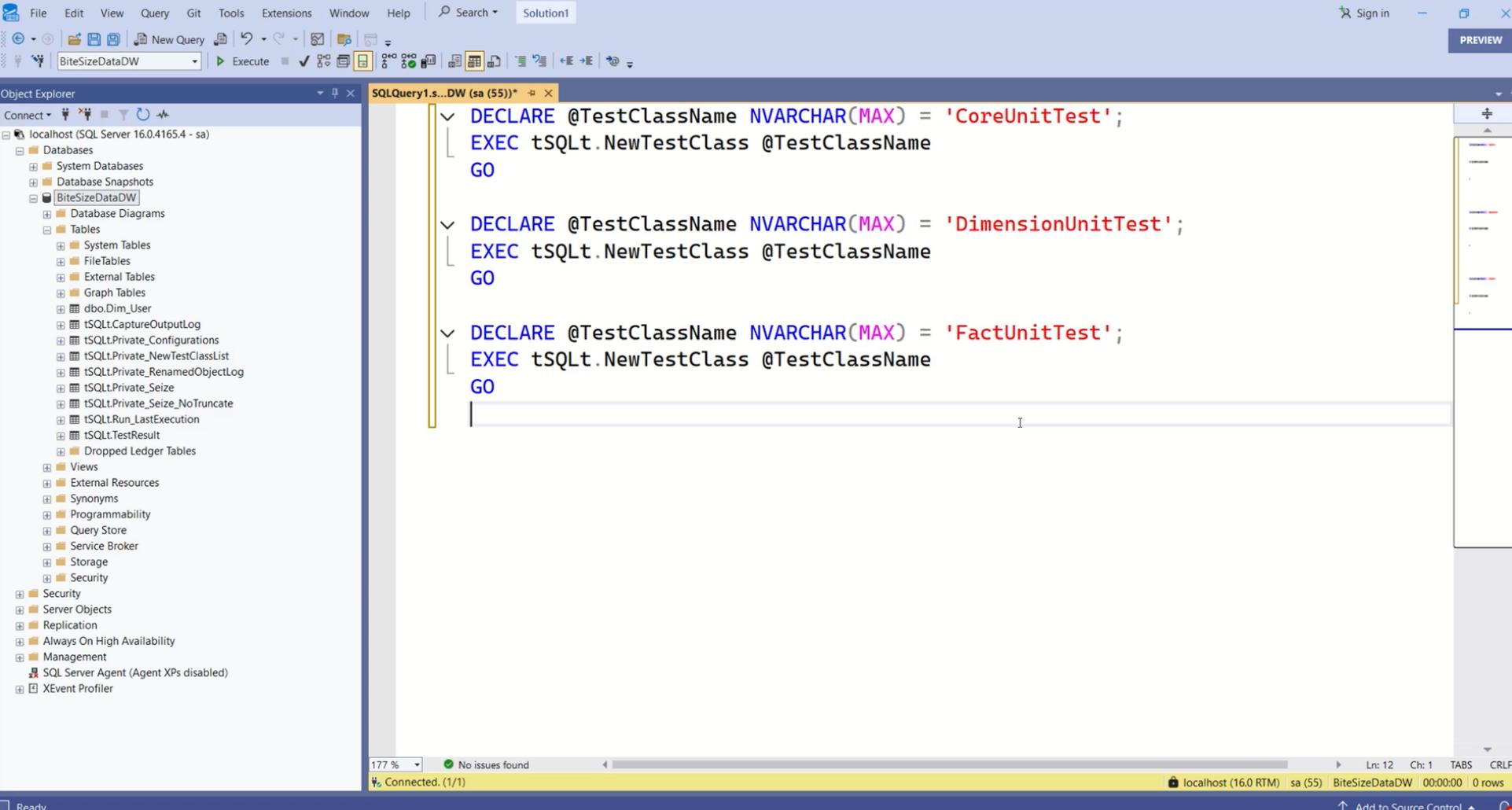
```
DECLARE @TestClassName NVARCHAR(MAX) =  
'CoreUnitTest';
```

```
EXEC tSQLt.NewTestClass @TestClassName
```

```
GO
```



Create a Test Class



The screenshot shows the SQL Server Management Studio (SSMS) interface. The Object Explorer on the left shows a connection to 'localhost (SQL Server 16.0.41654 - sa)' with the database 'BiteSizeDataDW' selected. The central query window displays the following T-SQL script:

```
DECLARE @TestClassName NVARCHAR(MAX) = 'CoreUnitTest';
EXEC tSQLt.NewTestClass @TestClassName
GO

DECLARE @TestClassName NVARCHAR(MAX) = 'DimensionUnitTest';
EXEC tSQLt.NewTestClass @TestClassName
GO

DECLARE @TestClassName NVARCHAR(MAX) = 'FactUnitTest';
EXEC tSQLt.NewTestClass @TestClassName
GO
```

The script uses the `tSQLt.NewTestClass` stored procedure to create three new test classes named 'CoreUnitTest', 'DimensionUnitTest', and 'FactUnitTest' respectively. The SSMS interface includes a status bar at the bottom indicating 'Connected. (1/1)' and connection details.



Testing Different Database Objects

Functions

Stored Procedures

Triggers

Values



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Create a Unit Test

Core Unit Test Class

→ Check if Dimension Table Exists



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

What makes a Unit Test?

Assemble

Act

Assert



DATA-CUISINE.COM
BITE SIZE DATA TALES



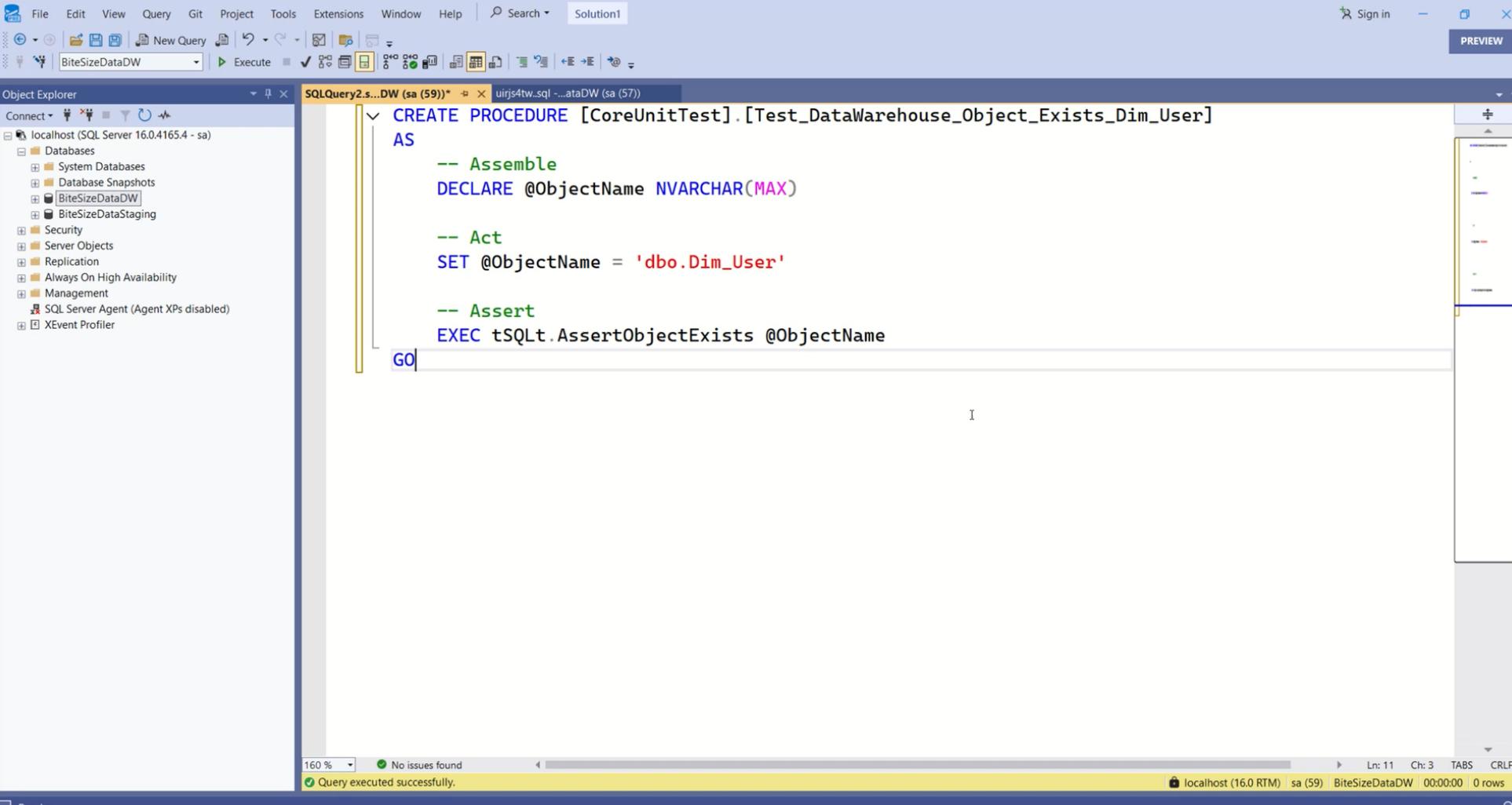
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Check if Dimension Table Exists



The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure under "localhost (SQL Server 16.0.4165.4 - sa)".
- SQLQuery2.s...DW (sa (59))*** is the active query window.
- Script:**

```
CREATE PROCEDURE [CoreUnitTest].[Test_DataWarehouse_Object_Exists_Dim_User]
AS
    -- Assemble
    DECLARE @ObjectName NVARCHAR(MAX)

    -- Act
    SET @ObjectName = 'dbo.Dim_User'

    -- Assert
    EXEC tSQLt.AssertObjectExists @ObjectName
GO
```
- Status Bar:** Shows "No issues found" and "Query executed successfully."
- Bottom Status:** "Ready", "160 %", "localhost (16.0 RTM) | sa (59) | BiteSizeDataDW | 00:00:00 | 0 rows", "Ln: 11 Ch: 3 TABS CRLF".



Which tSQLt did we use?

`tSQLt.AssertObjectExists`



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Execute Unit Tests

```
EXEC tSQLt.RunAll
```

```
(1 row affected)
```

```
+-----+  
| Test Execution Summary |  
+-----+
```

No	Test Case Name	Dur(ms)	Result
1	[CoreUnitTest].[Test_DataWarehouse_Object_Exists_Dim_User]	26	Success

```
Test Case Summary: 1 test case(s) executed, 1 succeeded, 0 skipped, 0 failed, 0 errored.
```



Execute Unit Tests

```
EXEC tSQLt.RunAll
```

```
DECLARE @TestName NVARCHAR(MAX) =  
'CoreUnitTest'
```

```
EXEC tSQLt.Run @TestName
```



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Create another Unit Test

Core Unit Test Class

→ Check Database Collation vs Server Collation



DATA-CUISINE.COM
BITE SIZE DATA TALES



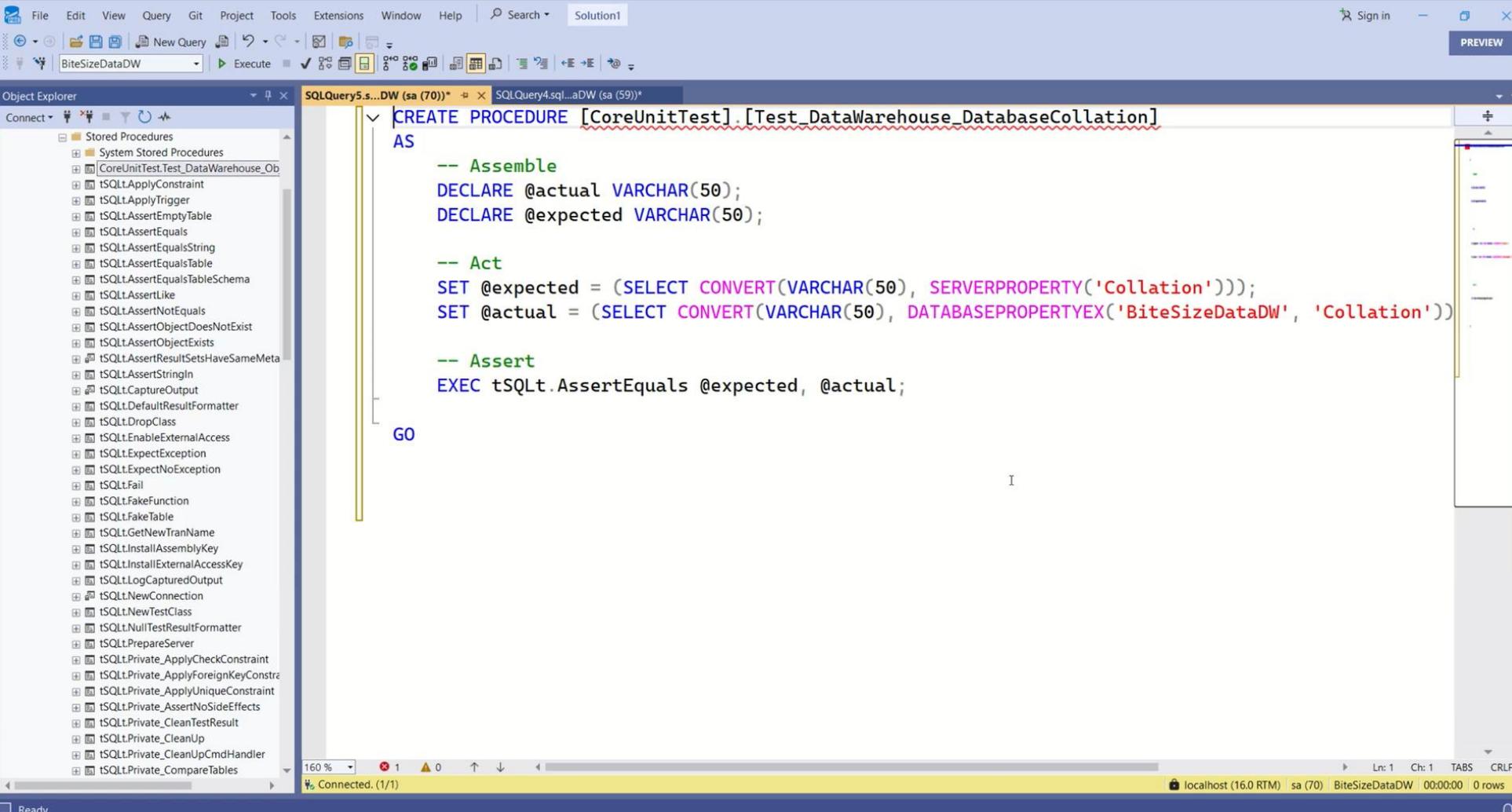
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Check Database Collation vs Server Collation



The screenshot shows the Object Explorer on the left with a list of stored procedures under the 'Stored Procedures' node. The central pane displays a T-SQL script for creating a stored procedure named `[CoreUnitTest].[Test_DataWarehouse_DatabaseCollation]`. The script uses the `tSQLt` library for unit testing. It defines variables `@actual` and `@expected` as `VARCHAR(50)`. It then selects the expected collation from `SERVERPROPERTY('Collation')` and the actual collation from `DATABASEPROPERTYEX('BiteSizeDataDW', 'Collation')`. Finally, it executes the `tSQLt.AssertEquals` method to compare the two values. The script concludes with a `GO` statement.

```
CREATE PROCEDURE [CoreUnitTest].[Test_DataWarehouse_DatabaseCollation]
AS
    -- Assemble
    DECLARE @actual VARCHAR(50);
    DECLARE @expected VARCHAR(50);

    -- Act
    SET @expected = (SELECT CONVERT(VARCHAR(50), SERVERPROPERTY('Collation')));
    SET @actual = (SELECT CONVERT(VARCHAR(50), DATABASEPROPERTYEX('BiteSizeDataDW', 'Collation')));

    -- Assert
    EXEC tSQLt.AssertEquals @expected, @actual;

GO
```



Testing a Dimension Load

Target Database: BiteSizeDataDW

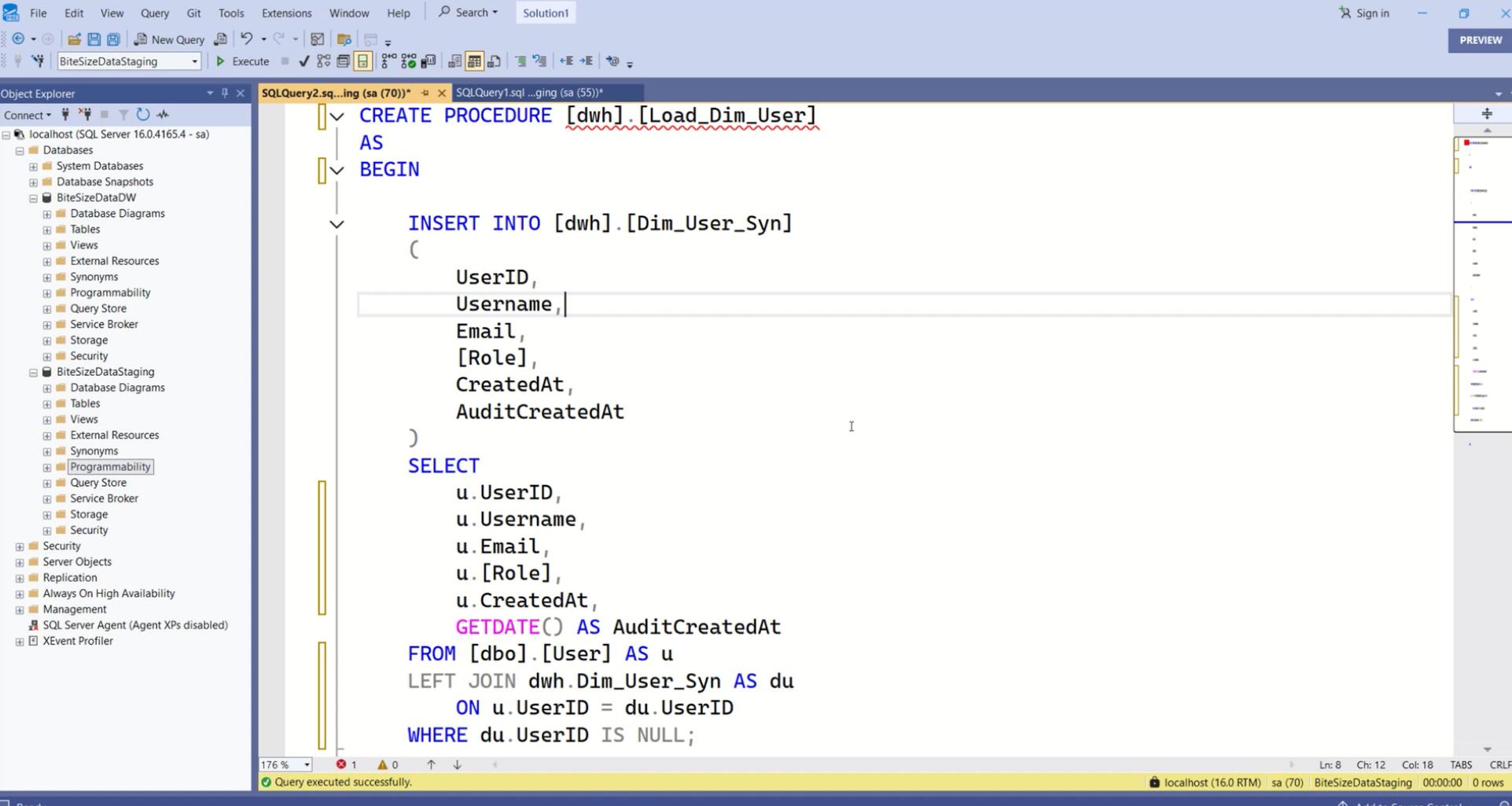
Source Database: BiteSizeDataStaging

Stored Procedure: dwh.Load_Dim_User

→ Testing cross database functionality!



Testing a Dimension Load



The screenshot shows the Object Explorer and a query editor in SQL Server Management Studio. The Object Explorer displays the database structure, including databases like localhost (sa), BitSizeDataDW, and BitSizeDataStaging, along with their tables, views, and other objects. The query editor contains a T-SQL script for creating a stored procedure named [dwh].[Load_Dim_User]. The script uses an INSERT INTO statement to insert data into the [dwh].[Dim_User_Syn] table, selecting columns from the [dbo].[User] table. It includes a SELECT statement to map columns, a LEFT JOIN to update the synonym table, and a WHERE clause to identify new users. The status bar at the bottom indicates the query was executed successfully.

```
CREATE PROCEDURE [dwh].[Load_Dim_User]
AS
BEGIN

    INSERT INTO [dwh].[Dim_User_Syn]
    (
        UserID,
        Username,
        Email,
        [Role],
        CreatedAt,
        AuditCreatedAt
    )
    SELECT
        u.UserID,
        u.Username,
        u.Email,
        u.[Role],
        u.CreatedAt,
        GETDATE() AS AuditCreatedAt
    FROM [dbo].[User] AS u
    LEFT JOIN dwh.Dim_User_Syn AS du
        ON u.UserID = du.UserID
    WHERE du.UserID IS NULL;
```



Which tSQLt did we use?

tSQLt.FakeTable → Dependency Isolation

tSQLt.AssertEqualsTable → Assertion



Dependency Isolation

ApplyConstraint

ApplyTrigger

FakeTable

FakeFunction

...

Assertions

AssertObjectExists

AssertEquals

AssertEqualsTable

...

AssertObjectDoesNotExist

AssertNotEquals

AssertLike



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Testing a Function (1/2)

Converting \$ → € based on a date

→ Will be used in Dimension load in the end



DATA-CUISINE.COM
BITE SIZE DATA TALES



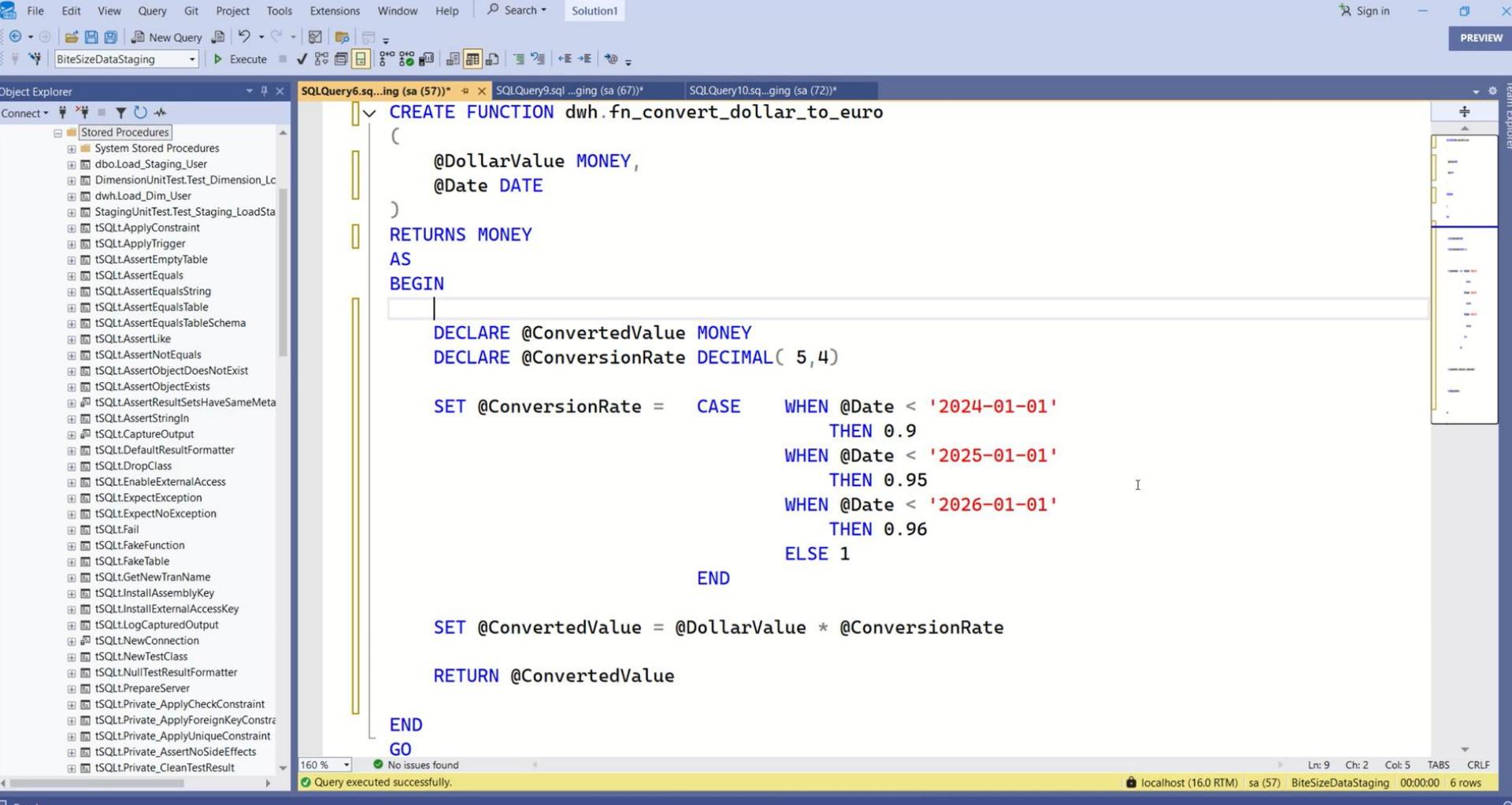
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Testing a Function



```
CREATE FUNCTION dwh.fn_convert_dollar_to_euro
(
    @DollarValue MONEY,
    @Date DATE
)
RETURNS MONEY
AS
BEGIN
    DECLARE @ConvertedValue MONEY
    DECLARE @ConversionRate DECIMAL( 5,4)

    SET @ConversionRate = CASE WHEN @Date < '2024-01-01'
                                THEN 0.9
                            WHEN @Date < '2025-01-01'
                                THEN 0.95
                            WHEN @Date < '2026-01-01'
                                THEN 0.96
                            ELSE 1
                           END

    SET @ConvertedValue = @DollarValue * @ConversionRate

    RETURN @ConvertedValue
END
GO
```



Getting tSQLt Test Results (1/2)

```
SELECT *
FROM tSQLt.TestResult
```



DATA-CUISINE.COM
BITE SIZE DATA TALES



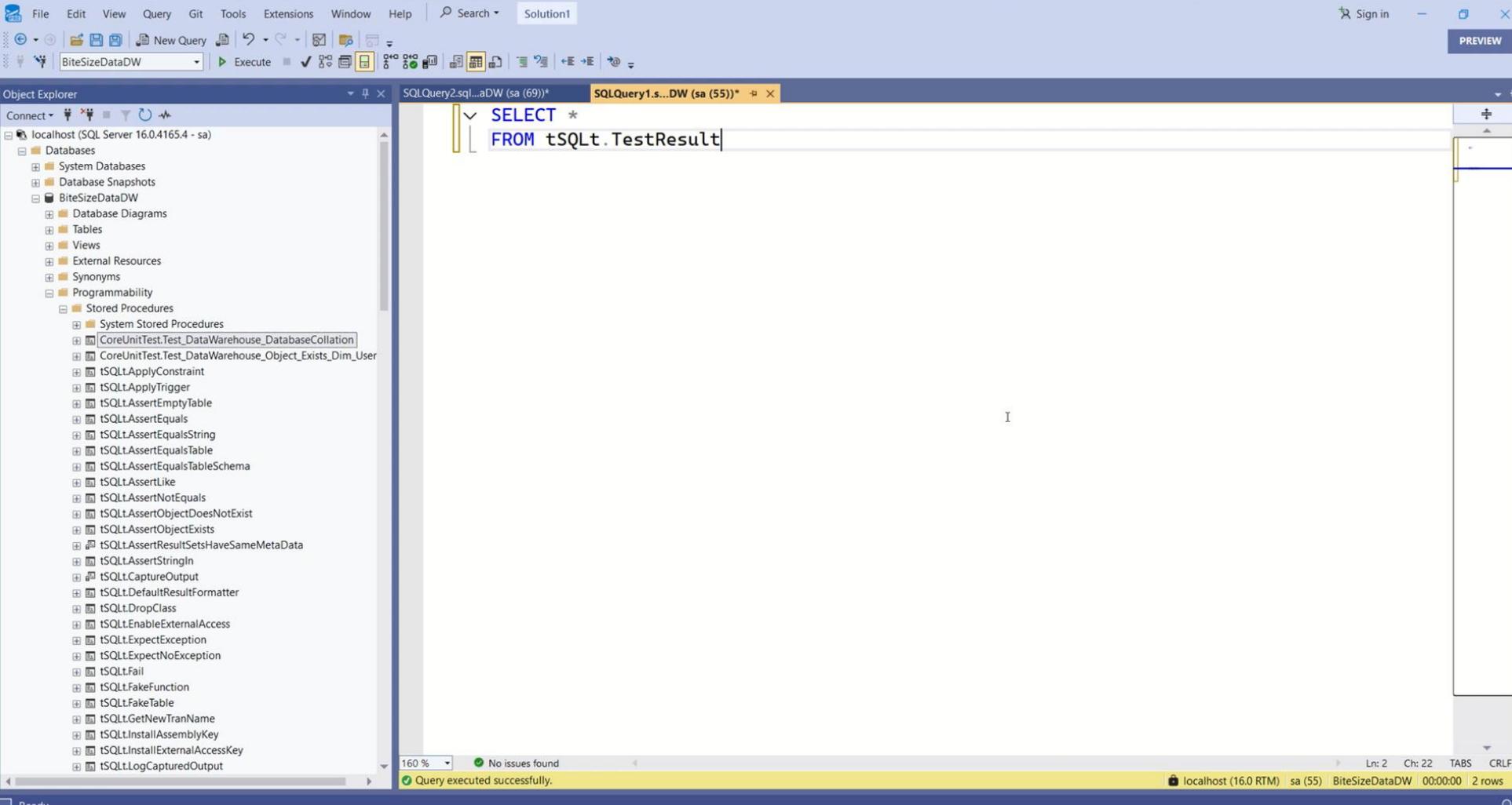
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Getting tSQLt Test Results (1/2)



The screenshot shows the SSMS interface. The Object Explorer on the left lists the database structure for 'BiteSizeDataDW'. The central query window contains the following T-SQL code:

```
SELECT *  
FROM tSQLt.TestResult
```

The status bar at the bottom indicates 'Query executed successfully.'



Getting tSQLt Test Results (2/2)

```
EXEC [tSQLt].[XmlResultFormatter]
```



DATA-CUISINE.COM
BITE SIZE DATA TALES



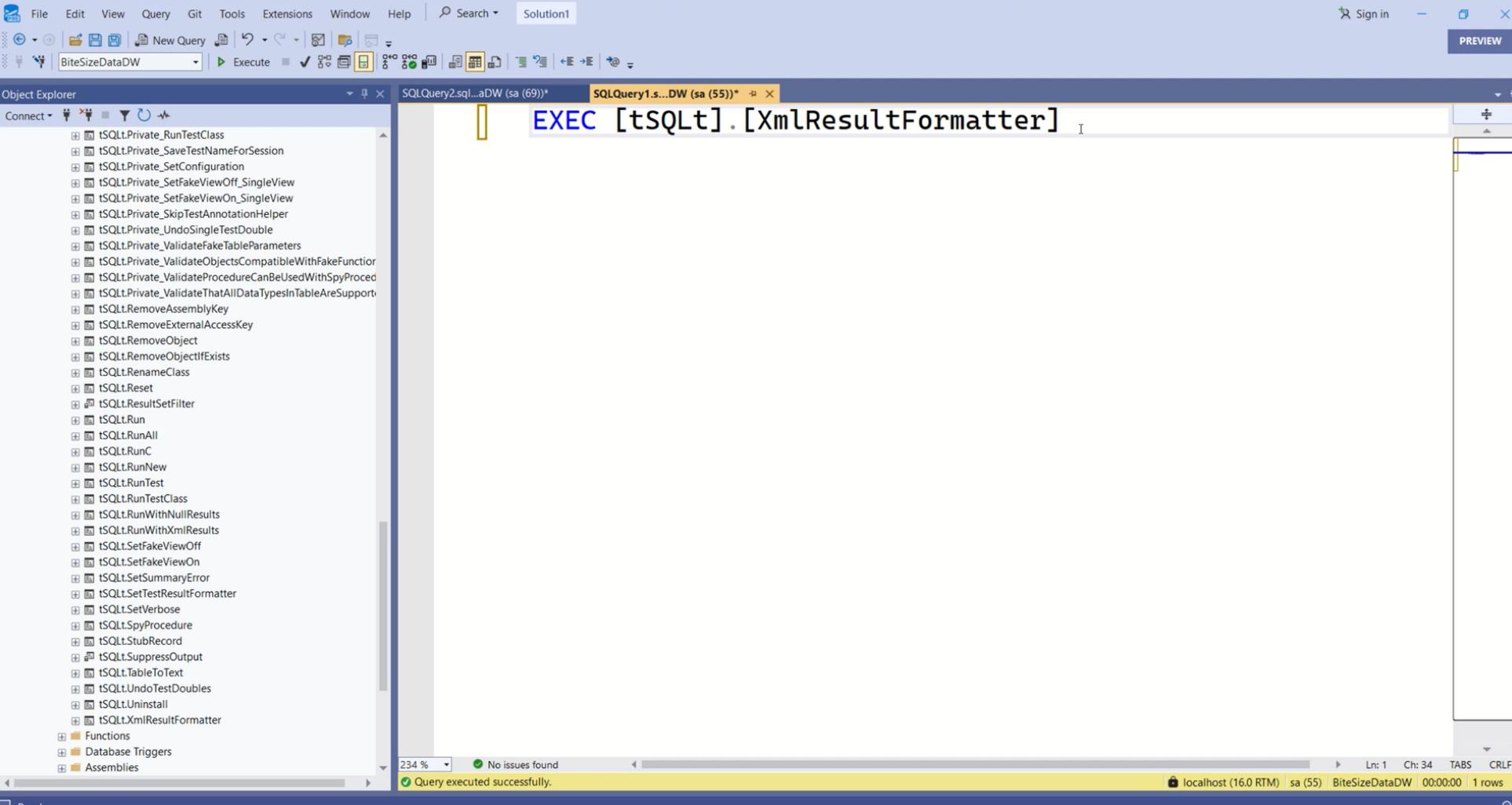
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Getting tSQLt Test Results (2/2)



The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows a tree view of objects under the schema [tSQLt].
- SQL Query Editor:** Displays the command: `EXEC [tSQLt].[XmlResultFormatter]`.
- Status Bar:** Shows "Query executed successfully."
- Bottom Right:** Shows connection information: "localhost (16.0 RTM) | sa (55) | BiteSizeDataDW | 00:00:00 | 1 rows".



Automation



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Automation

Azure DevOps

DbaTools

PowerShell

Database Projects



DATA-CUISINE.COM
BITE SIZE DATA TALES



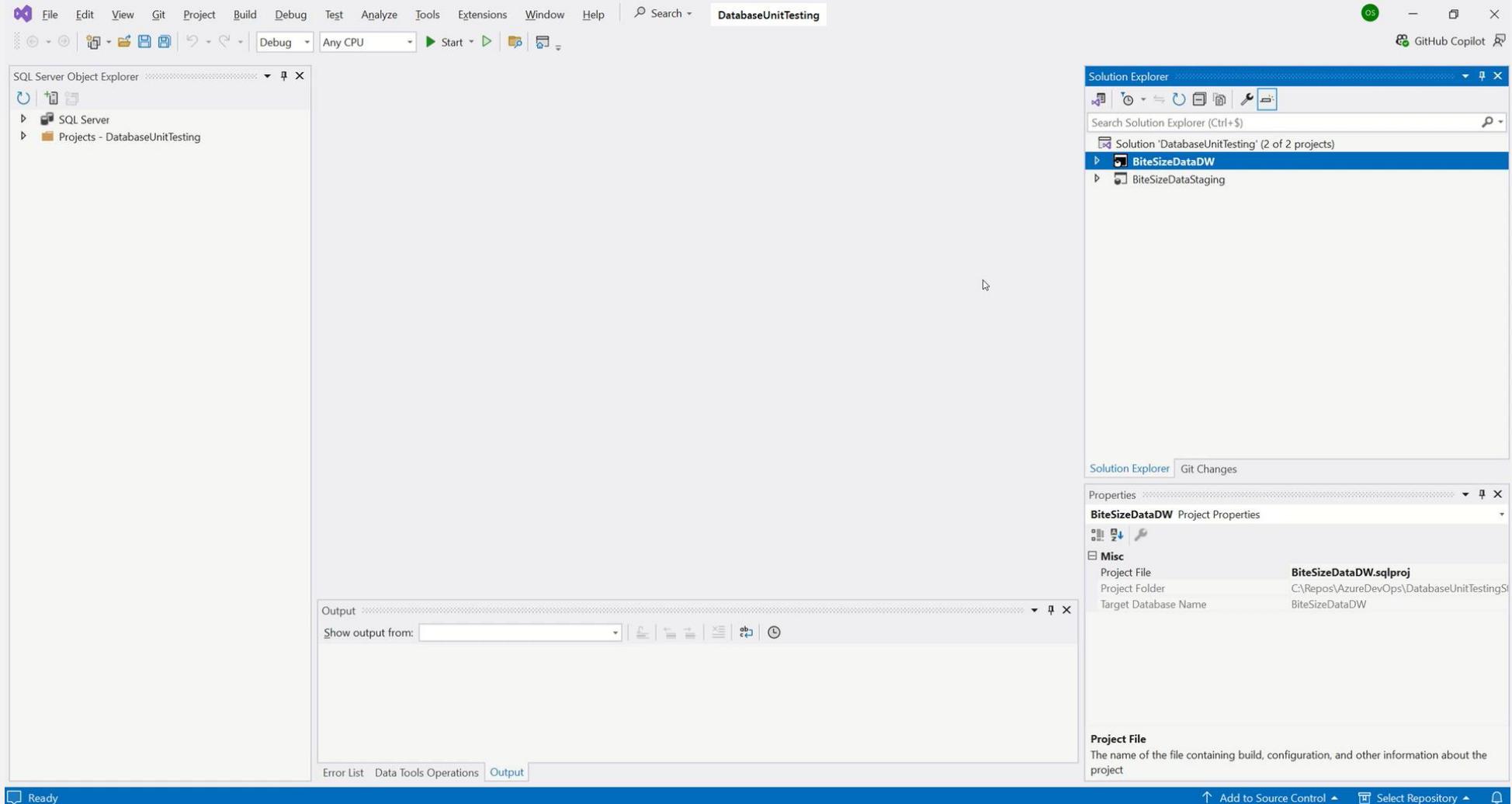
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Setting up the Database Project(s)



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Setting up the Database Project(s)

→BiteSizeDataDW_UnitTesting

→BiteSizeDataStaging_UnitTesting

→tSQLt_Framework



DATA-CUISINE.COM
BITE SIZE DATA TALES



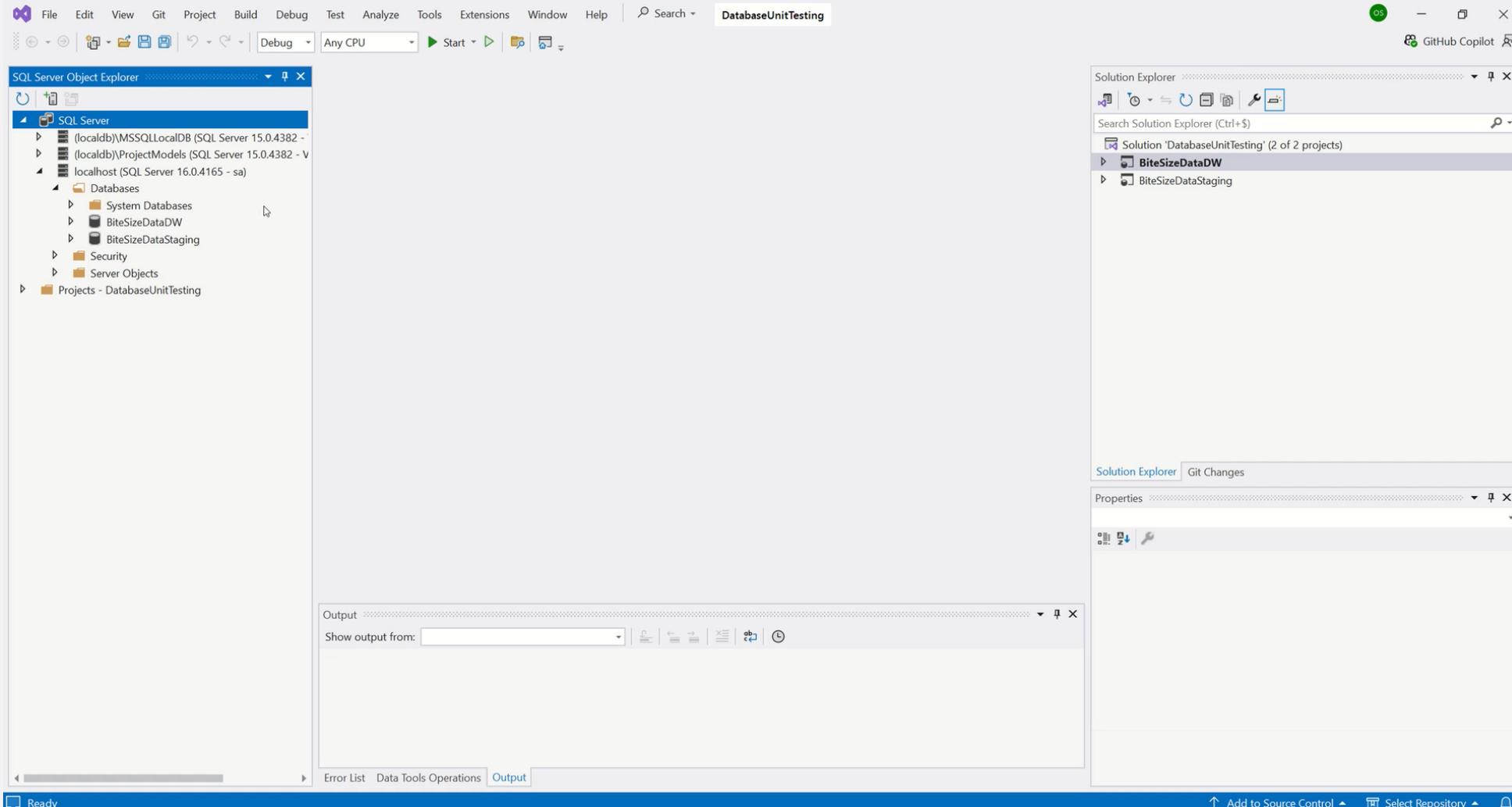
/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

tSQLt_Framework Database Project



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

tSQLt_Framework Database Project

Only for reference

Contains all reference objects



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

BiteSizeDataDW_UnitTesting Database Project

The screenshot shows the Microsoft Visual Studio interface with the following components:

- Solution Explorer:** Shows three projects: BitSizeDataDW, BitSizeDataStaging, and tSQLt_Framework.
- SQL Server Object Explorer:** Displays the structure of the BitSizeDataDW database on the localhost server.
- Output Window:** Shows the build logs for the tSQLt_Framework project, indicating a successful build.



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Current Setup Warnings

1 way to setup (there are other ways too)

For every new version → changes to make

Alternative → Reference the .dacpac

Alternative → Post Deployment or separate scripts



CI-Unit-Testing Pipeline

The screenshot shows the Azure DevOps Pipelines interface. The left sidebar navigation bar is visible, with the 'Pipelines' section currently selected. The main content area displays a table titled 'Recently run pipelines'.

Pipeline	Last run	
CD-Unit-Testing	#20250116.27 • Update CD-Unit-Testing.yml for Azure Pipelines Individual CI for master	6m ago 1m 49s
CI-Unit-Testing	#20250116.11 • Update CD-Unit-Testing.yml for Azure Pipelines Individual CI for master	34m ago 1m 2s



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

CD-Unit-Testing Pipeline

The screenshot shows the Azure DevOps Pipelines interface for the project "Unit Testing for Database D...". The left sidebar is open, showing the "Pipelines" section selected. The main area displays the "Recently run pipelines" table.

Pipeline	Last run
CD-Unit-Testing	#20250116.27 • Update CD-Unit-Testing.yml for Azure Pipelines ⌚ Individual CI for ⚡ master 🕒 7m ago 🕒 1m 49s
CI-Unit-Testing	#20250116.11 • Update CD-Unit-Testing.yml for Azure Pipelines ⌚ Individual CI for ⚡ master 🕒 35m ago 🕒 1m 2s



CD-Unit-Testing Pipeline

The screenshot shows the Azure DevOps Pipelines interface. The left sidebar navigation bar is visible, with the 'Pipelines' item selected. The main content area displays a table titled 'Recently run pipelines'. The table has two columns: 'Pipeline' and 'Last run'. It lists two recent runs:

Pipeline	Last run
CD-Unit-Testing	#20250116.27 • Update CD-Unit-Testing.yml for Azure Pipelines ⌚ Individual CI for ⚡ master 🕒 13m ago 🕒 1m 49s
CI-Unit-Testing	#20250116.11 • Update CD-Unit-Testing.yml for Azure Pipelines ⌚ Individual CI for ⚡ master 🕒 41m ago 🕒 1m 2s



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Test Driven Development



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

TDD Concepts

Write Unit Tests before starting development

Write the minimal required code to pass

Functionality testing from the start



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

TDD Benefits & Challenges

- + Requirements More Clear - Increase Code Volume
- + Less debugging in the end - Maintenance overhead
- + Documentation - Time-consuming



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Need assistance?

Shoutout to Sander Stad!

Generate your Unit Tests:

<https://github.com/sanderstad/PStSQLtTestGenerator>

→ Works for: Database Collation, Objects Existence, Function Parameters, Stored Procedure Parameters, Table Columns, View Columns



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Summary



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

Summary

Write Basic Unit Tests ✓

Structure Unit Tests in a Database Project ✓

Publish Unit Tests ✓

Execute Unit Tests Automatically ✓

Up Next: Code Analysis?



DATA-CUISINE.COM
BITE SIZE DATA TALES

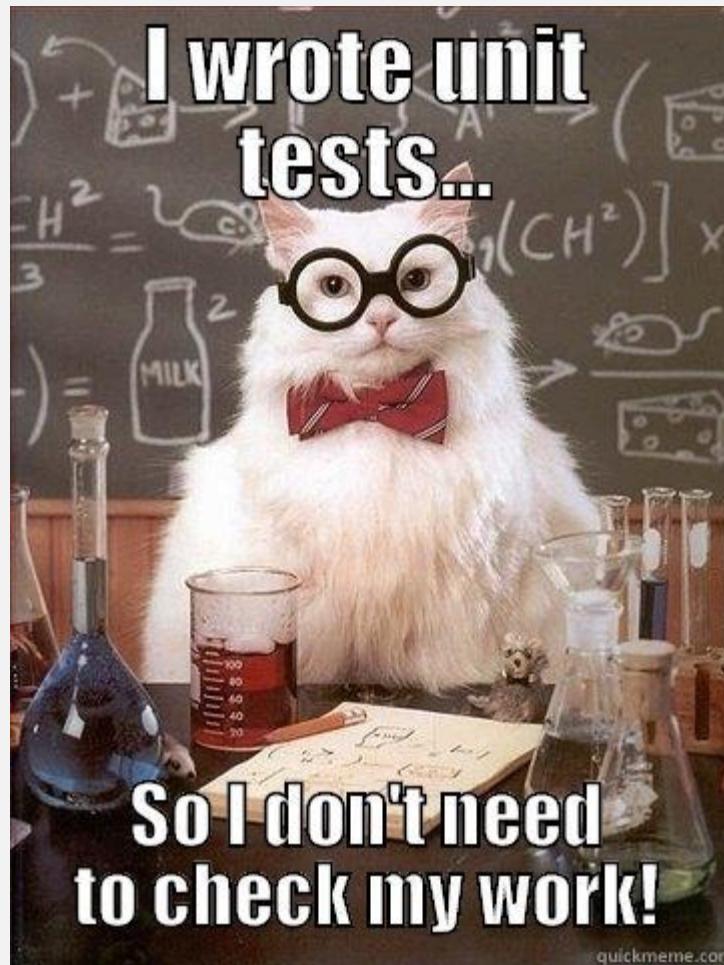


/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social

About me

Olivier Van Steenlandt



Expert @ datashift



Core Member @ dataMinds.be



Speaking / Blogging



Running / Cycling / Swimming



DATA-CUISINE.COM
BITE SIZE DATA TALES



/in/oliviervansteenlandt



@Oli_VSteenlandt

@oliviervs.bsky.social