Biml Tips and Tricks

Not just for SSIS Packages!

Cathrine Wilhelmsen

SQLGrillen · June 22nd 2018

DATAmasterminds

Session Description

"Wait, what? Biml is not just for generating SSIS packages?"

Absolutely not! Come and see how you can use Biml (Business Intelligence Markup Language) to save time and speed up other Data Warehouse development tasks. You can generate complex T-SQL statements with Biml instead of using dynamic SQL, create test data and populate static dimensions, and even compare tables and views across multiple servers and databases.

Don't Repeat Yourself, start automating those boring, manual tasks today!

Cathrine Wilhelmsen

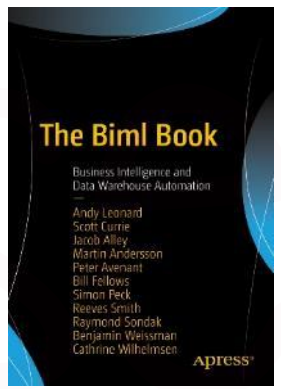
Data Warehouse & Business Intelligence Consultant

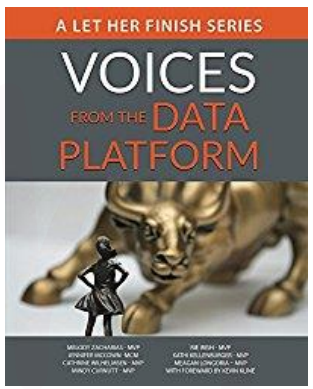
@cathrinew

cathrinew.net

Microsoft® Most Valuable Professional

BimlHero CERTIFIED EXPERT



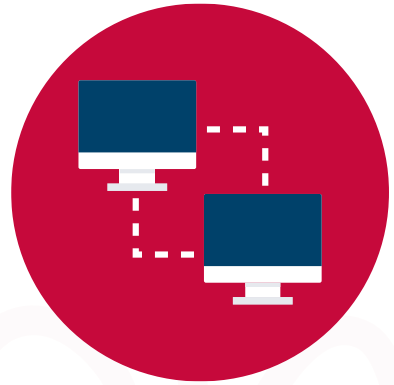


...the next 60 minutes...

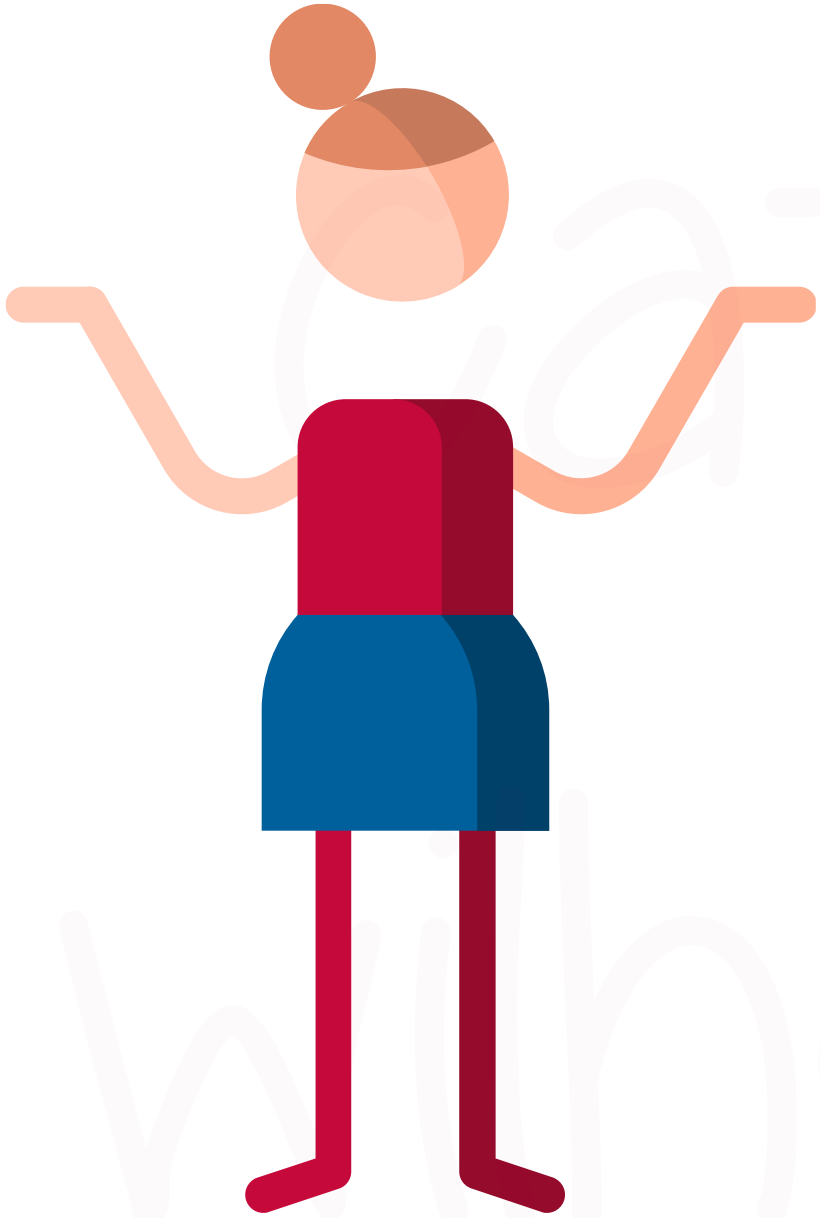
T-SQL

Test Data

Dimensions



Comparing



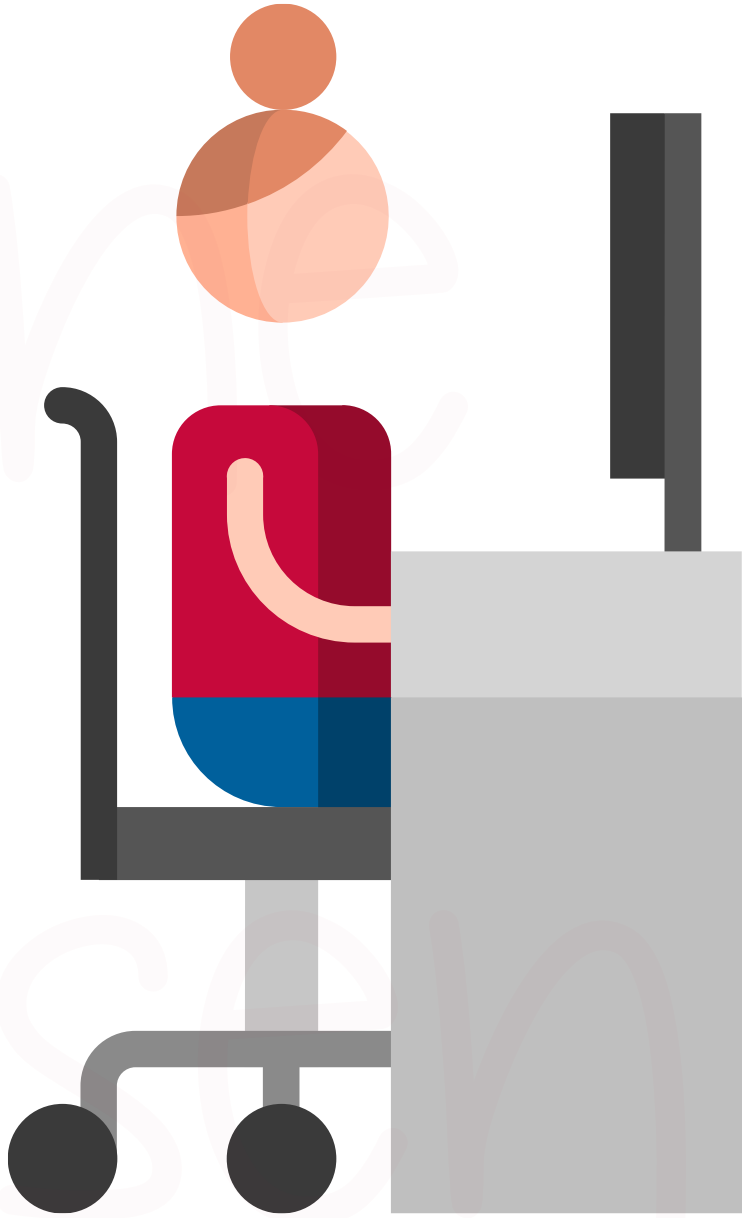
Wait...

Can't I use

<something else>?

Of course!

But Biml works well with source metadata

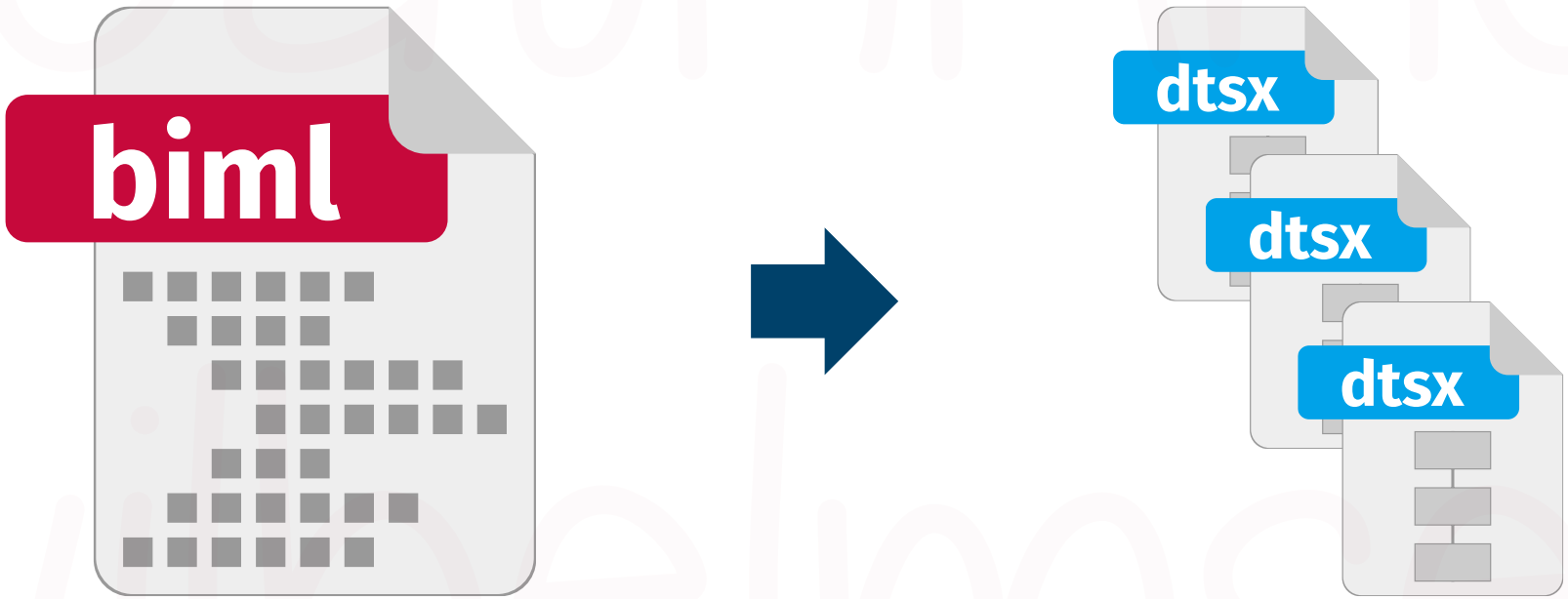


And...

Biml is fun!

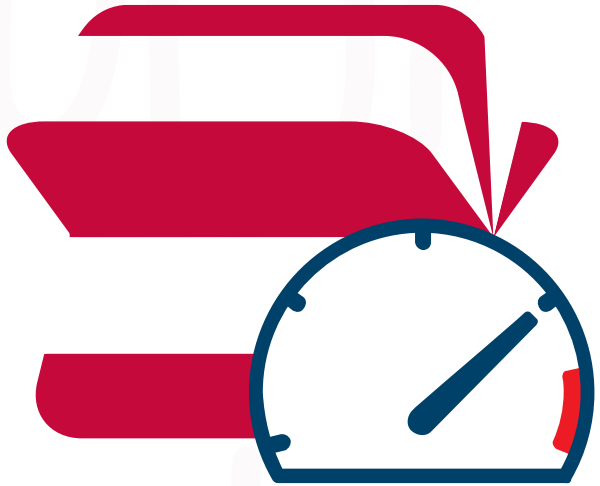
But... Biml? How?

Traditional Biml



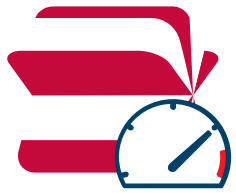
~~Crazy~~ Fun Biml

What do you need?



BimlExpress

The power is in the...



BimlExpress

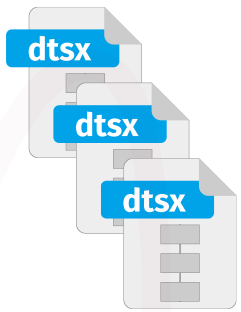
Preview Pane

Traditional BimlScript

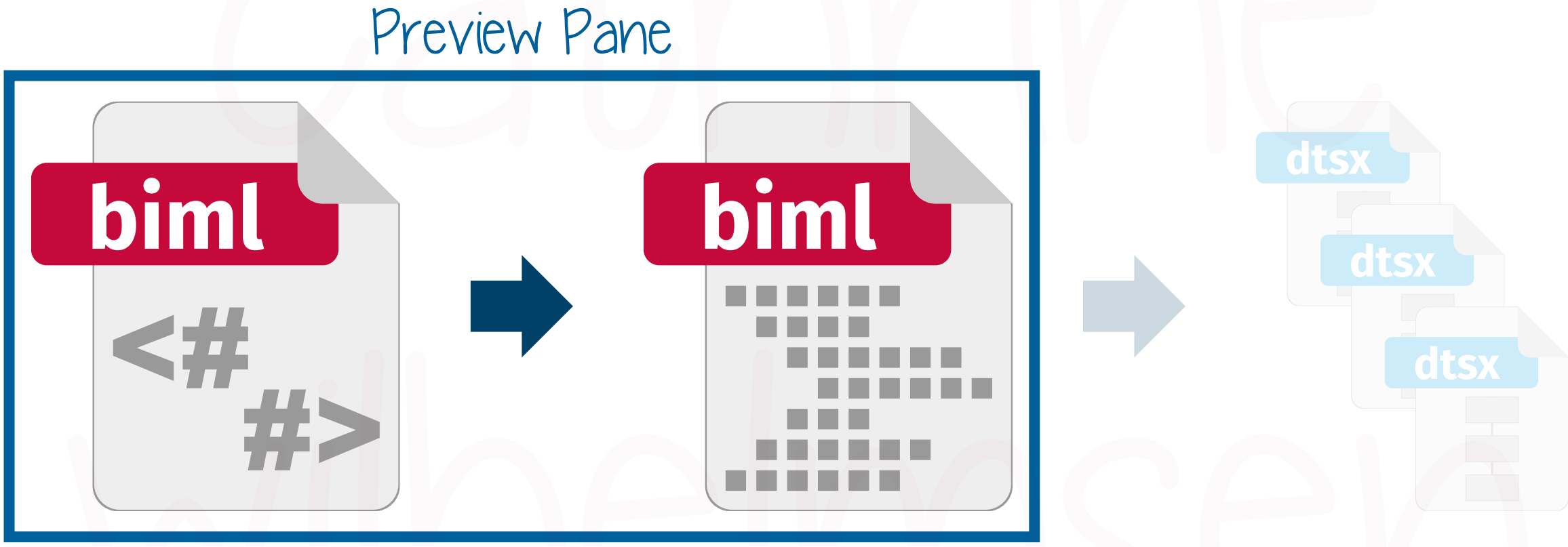
Traditional BimlScript

```
<# foreach (var table in RootNode.Tables) { #>
  <Package Name="Load_<#=table.Name#>" />
<# } #>
```

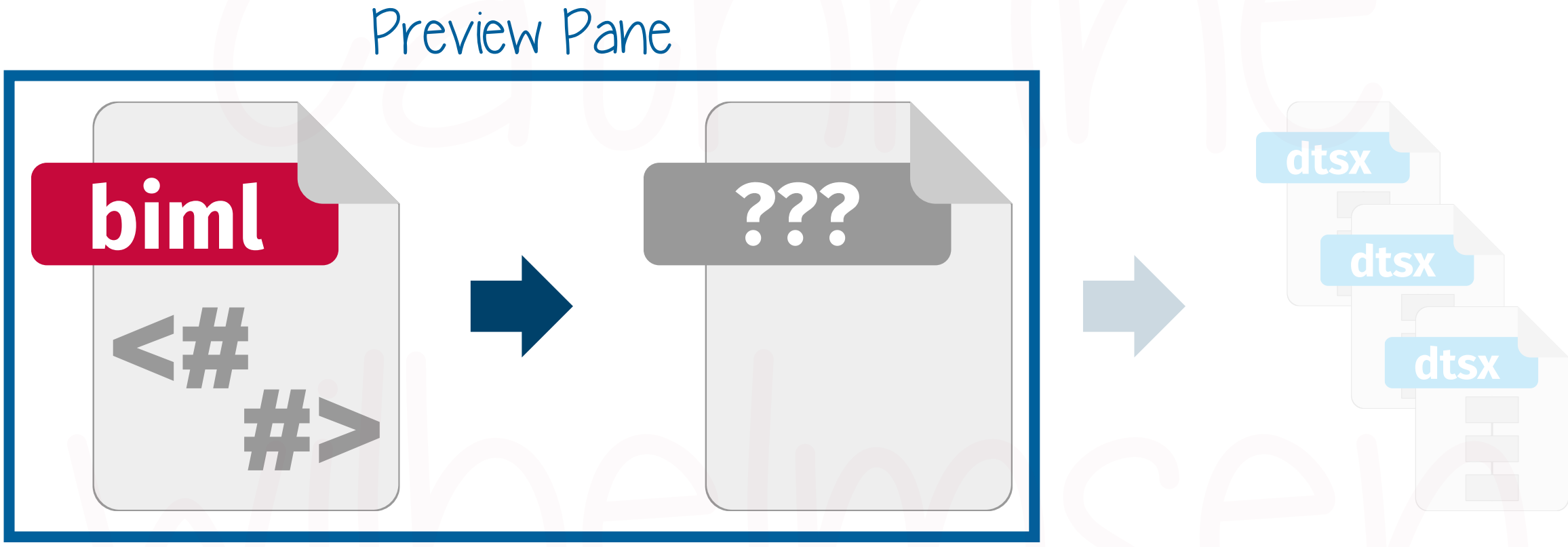
```
<Package Name="Load_Customer" />
<Package Name="Load_Product" />
<Package Name="Load_Sales" />
```



BimlScript to Biml



BimlScript to... ???



~~Crazy~~ Fun BimlScript

```
<# foreach (var table in RootNode.Tables) { #>
  Yay, a package! <#=table.Name#> :)
<# } #>
```

```
Yay, a package! Load_Customer :)
Yay, a package! Load_Product :)
Yay, a package! Load_Sales :)
```

T-SQL from Biml

Built-in

T-SQL from Biml

Built-in T-SQL from Biml

Functional T-SQL

```
GetSelectSql

GetDropAndCreateDdl

GetTableSql
```

SSMS Snippets

```
GetInsertSql

GetUpdateSql

GetDeleteSql
```

Functional T-SQL: **GetSelectSql**

```
SELECT
    [Column1]
   ,[Column2]
FROM [Schema].[Table]
```

Functional T-SQL: **GetDropAndCreateDdl**

```
IF EXISTS (...)
  DROP TABLE [Schema].[Table]
GO

CREATE TABLE [Schema].[Table] (
   [Column1] int IDENTITY(1,1) NOT NULL
  ,[Column2] nvarchar(50) NOT NULL
-- Constraints
)
```

Functional T-SQL: **GetTableSql**

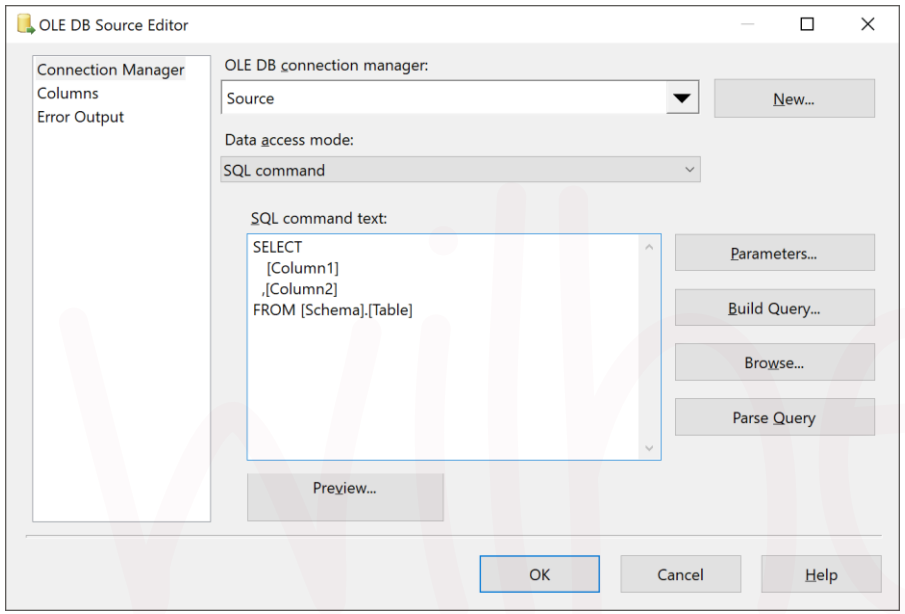
```
IF EXISTS (...)
  DROP TABLE [Schema].[Table]
GO

CREATE TABLE [Schema].[Table] (
   [Column1] int IDENTITY(1,1) NOT NULL
  ,[Column2] nvarchar(50) NOT NULL
-- Constraints
)
```

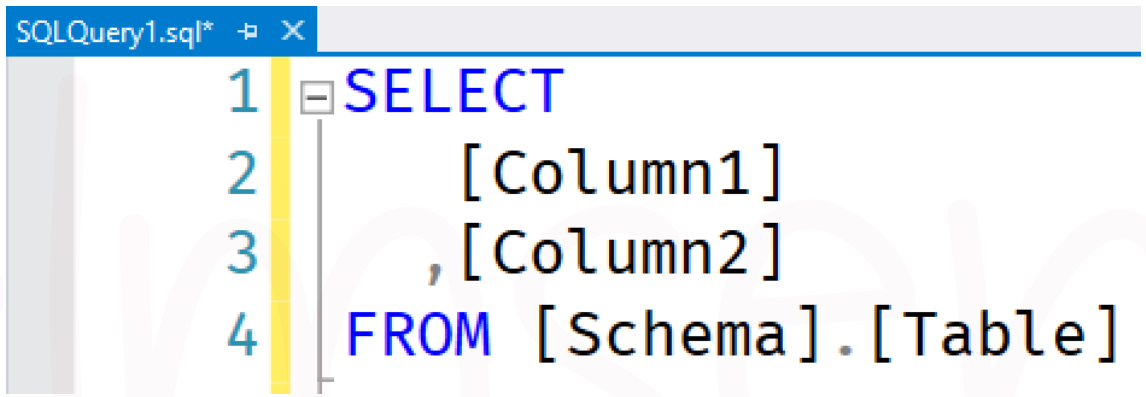
Yes, this is exactly the same as GetDropAndCreateDdl()

Functional T-SQL

Use directly in SSIS

Copy and run in SSMS





SSMS Snippets: **GetDeleteSql**

```
DELETE FROM [Schema].[Table]
WHERE <Search Conditions,,>
```

SSMS Snippets: **GetInsertSql**

```
INSERT INTO [Schema].[Table] (
    [Column1]
   ,[Column1]
) VALUES (
   <Column1,int,>,
   <Column2,nvarchar(50),>
)
```

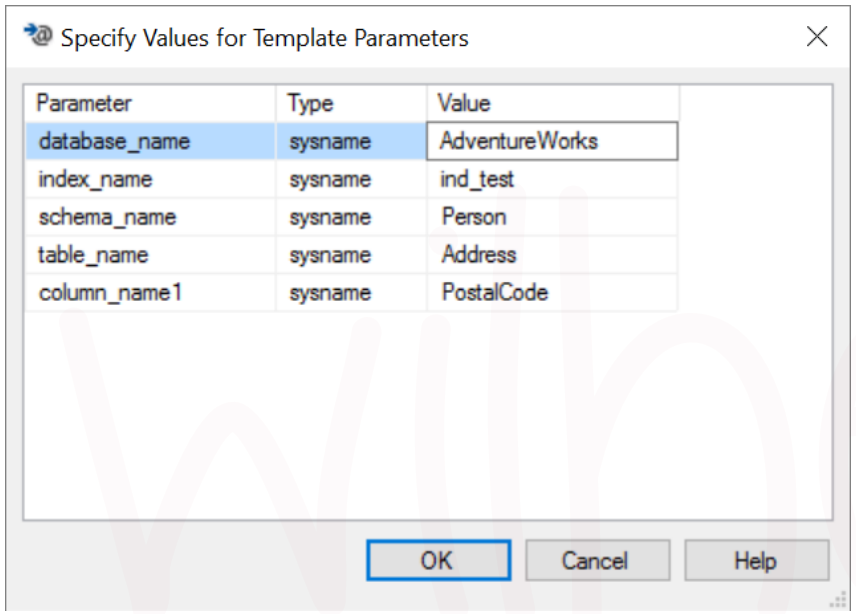
SSMS Snippets: **GetUpdateSql**

```
UPDATE [Schema].[Table] SET
   [Column1] = <Column1, int,>,
   [Column2] = <Column2, nvarchar(50),>
WHERE <Search Conditions,,>
```

SSMS Snippets

Copy and run in SSMS

Replace Template Parameters with actual values



Custom

T-SQL from Biml

Custom T-SQL from Biml

Table Metadata

```
<#=table.Name#>

<#=table.SchemaQualifiedName#>

SELECT *
```

Custom T-SQL from Biml

Column Methods

```
GetColumnList

GetColumnAssignmentList

GetColumnComparisonList
```

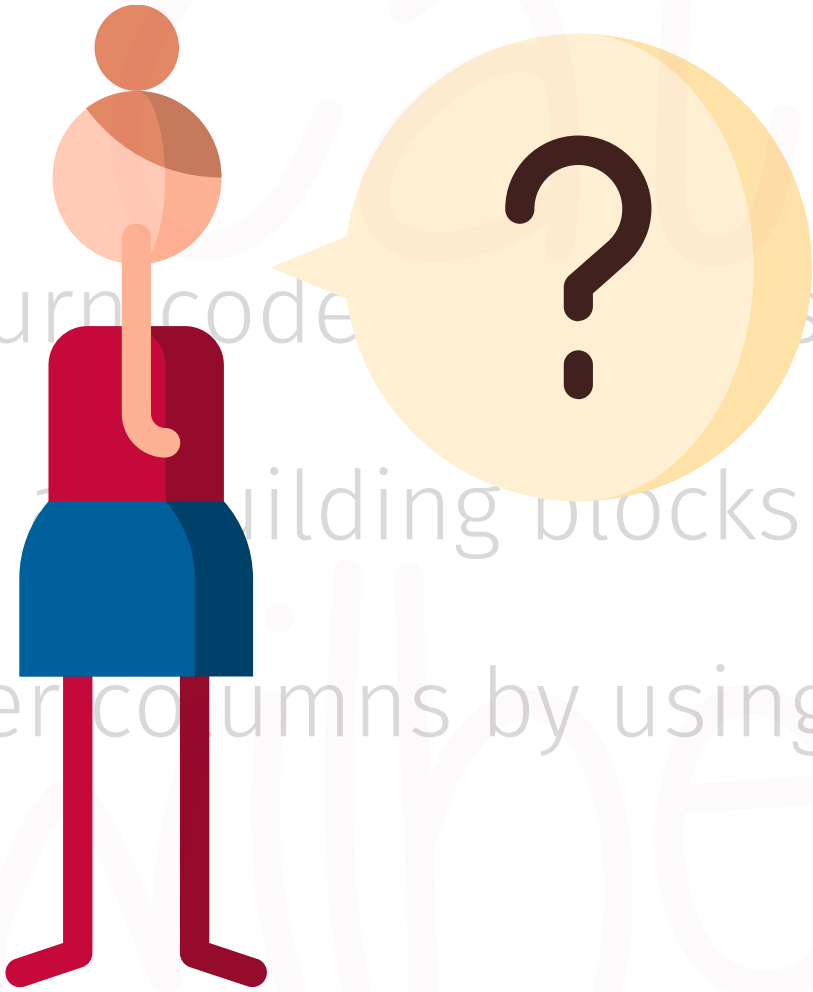
Column Methods

Return code fragments

Use as building blocks in custom T-SQL

Filter columns by using lambda expressions

Column Methods



Return codes

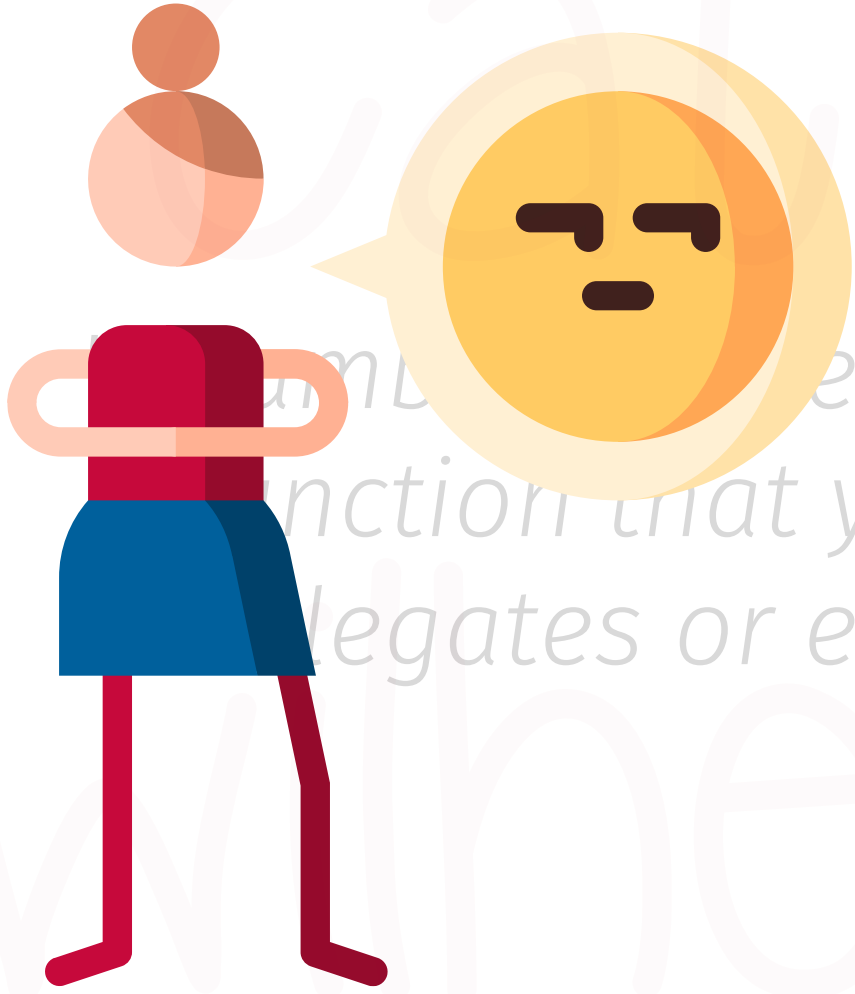
Use building blocks in custom T-SQL

Filter columns by using lambda expressions

Lambda Expressions

"A lambda expression is an anonymous function that you can use to create delegates or expression tree types"

Lambda Expressions



"A lambda expression is an anonymous function that you can use to create delegates or expression tree types"

Lambda Expressions

```
column => column.Name == "ColumnID"
```

Lambda Expressions

```
column => column.Name == "ColumnID"
```

The arrow is the lambda operator

Lambda Expressions

```
column => column.Name == "ColumnID"
```

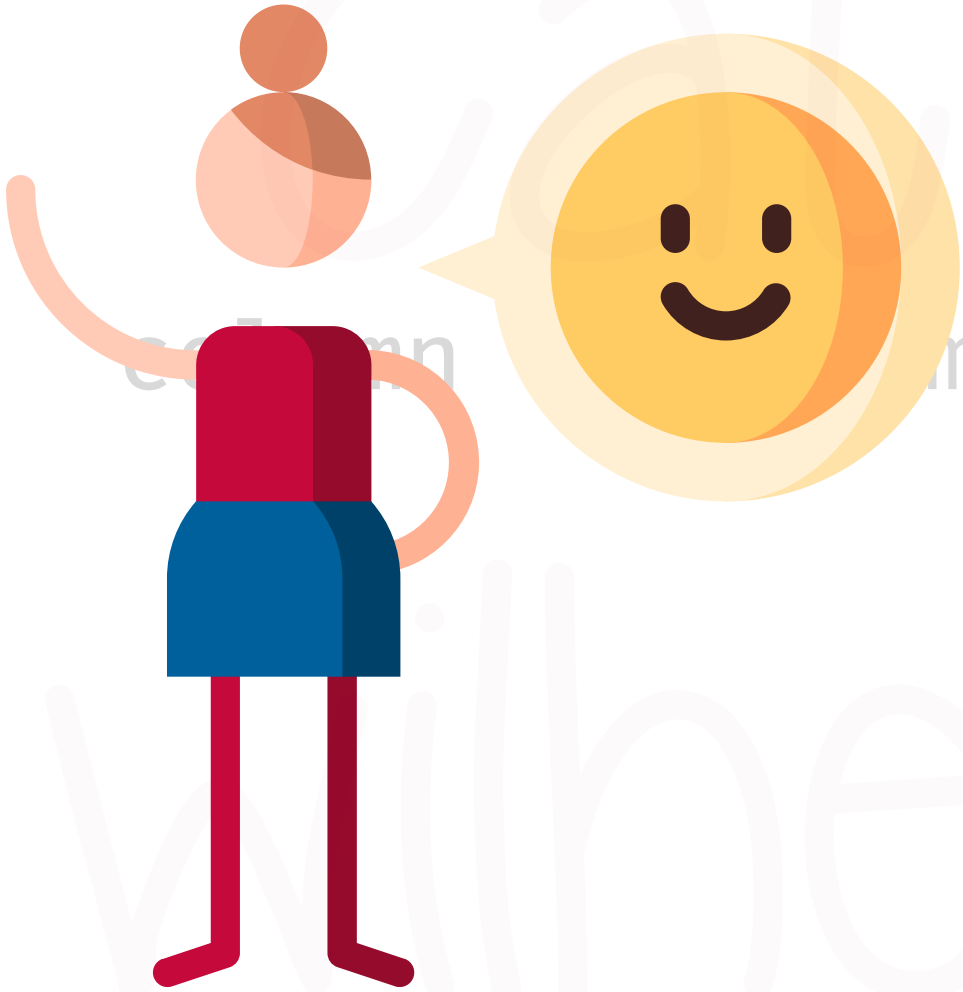
Input parameter is on the left side

Lambda Expressions

```
column => column.Name == "ColumnID"
```

Expression is on the right side

Lambda Expressions



```
column.Name == "ColumnID"
```

GetColumnList

Get columns for **SELECT**

```
[Column1], [Column2], [Column3]
```

GetColumnList

```
GetColumnList()

GetColumnList("src")

GetColumnList(c => c.Name == "ColumnID")

GetColumnList(c => c.DataType != DbType.Guid, "src")
```

GetColumnAssignmentList

Get columns for **UPDATE … SET**

```
[l].[Column1] = [r].[Column1]
,[l].[Column2] = [r].[Column2]
,[l].[Column3] = [r].[Column3]
```

GetColumnAssignmentList

```
GetColumnAssignmentList()

GetColumnAssignmentList("dst", "src")

GetColumnAssignmentList(c => !c.IsUsedInPrimaryKey)

GetColumnAssignmentList(c => !c.IsUsedInPrimaryKey, "dst", "src")
```

GetColumnComparisonList

Get columns for **JOIN ... ON**

```
[l].[Column1] = [r].[Column1]
AND [l].[Column2] = [r].[Column2]
AND [l].[Column3] = [r].[Column3]
```

GetColumnComparisonList

```
GetColumnComparisonList()

GetColumnComparisonList("!=")

GetColumnComparisonList("dst", "src")

GetColumnComparisonList(c => c.IsUsedInPrimaryKey)

GetColumnComparisonList(c => c.IsUsedInPrimaryKey, "dst", "src")

GetColumnComparisonList(c => c.IsUsedInPrimaryKey, "!=", "dst", "src")
```

DEMO

T-SQL from Biml

Test Data

Biml + Random()

Create Random Test Data

Random Class

Random()

Pseudo-random number generator

Random enough for practical purposes

Random(12345)

Always generates the same series of random numbers

Random Methods

Next()	Random number
Next(100)	Random number smaller than specified value
Next(1000, 2000)	Random number within specified range
NextDouble()	Random decimal number between 0.0 and 1.0

Biml Helper Method

```
<#+
  Random r = new Random();
  public string GetTestData(AstTableColumnBaseNode column) {
    switch (column.DataType) {
      case DbType.Int32 :
        return r.Next(Int32.MaxValue).ToString();
      case DbType.String :
        return "'" + GetRandomString(column.Length) + "'"
      default :
        return "NULL";
    }
  }
#>
```

Biml Helper Helper Method

```
<#+
  private string GetRandomString(int length) {
    var allowedChars = "ABC123";
    var randomChars = new char[length];
    for (int i = 0; i < length; i++) {
      randomChars[i] = allowedChars[ r.Next(allowedChars.Length) ];
    }
    return new String(randomCharacters);
  }
#>
```

Biml + Random()

Very limited functionality

No dependencies

Works out-of-the box

Biml + Bogus

Create Random and Specific Test Data

Bogus Project

Simple and sane fake data generator for .NET

https://github.com/bchavez/Bogus

Advanced functionality for custom objects

...or simple functionality for Biml hacks :)

Bogus API

Supports all data types plus built-in test data, including:

Address	Date	Name
Commerce	Finance	Person
Company	Internet	Phone
Database	Lorem	System

The Bogus Methods

```
<#@ assembly name="Bogus.dll" #>
<#@ import namespace="Bogus" #>

<# var f = new Faker(); #>
<# Person p = new Person(); #>

INSERT INTO dbo.Employee(FirstName, LastName, Birthday) VALUES
(
  '<#=p.FirstName#>',
  '<#=p.LastName#>',
  '<#=p.DateOfBirth.ToString("yyyyMMdd")#>'
)
```

Biml + Bogus()

Better functionality than Random()

Easier to understand, frequently updated

Requires additional files

Installing Bogus

Install via Nuget:

```
Install-Package Bogus
```

Or download latest release .zip:

https://github.com/bchavez/Bogus/releases

DEMO

Test Data with Random() and Bogus

Alternative (*Better*) Solutions

Mockaroo (free, but only 1000 rows)

https://mockaroo.com/

Redgate SQL Data Generator (licensed)

https://www.red-gate.com/products/sql-development/sql-data-generator/index

Static Dimensions

Static Dimensions (or any kind of static table)

SCD Type 0: Not changing

Lookup Tables

1. Create table definition
2. Add static source rows
3. Generate INSERT statements

Biml Static Sources

Part of `<Table>` objects

Defines rows to be inserted when table is created

- Unknown dimension members
- Date dimension
- Code tables

Biml Static Source

```
<Table Name="Table" SchemaName="Database.Schema">
    <Columns>
        <Column Name="Column1" DataType="Int32" />
        <Column Name="Column2" DataType="String" Length="10" />
    </Columns>
    <Sources>
        <StaticSource Name="TableRows">
            <Rows>
                <Row>
                    <ColumnValues>
                        <ColumnValue ColumnName="Column1" Value="-1" />
                        <ColumnValue ColumnName="Column2" Value="'Unknown'" />
                    </ColumnValues>
                </Row>
            </Rows>
        </StaticSource>
    </Sources>
</Table>
```

Biml Static Source

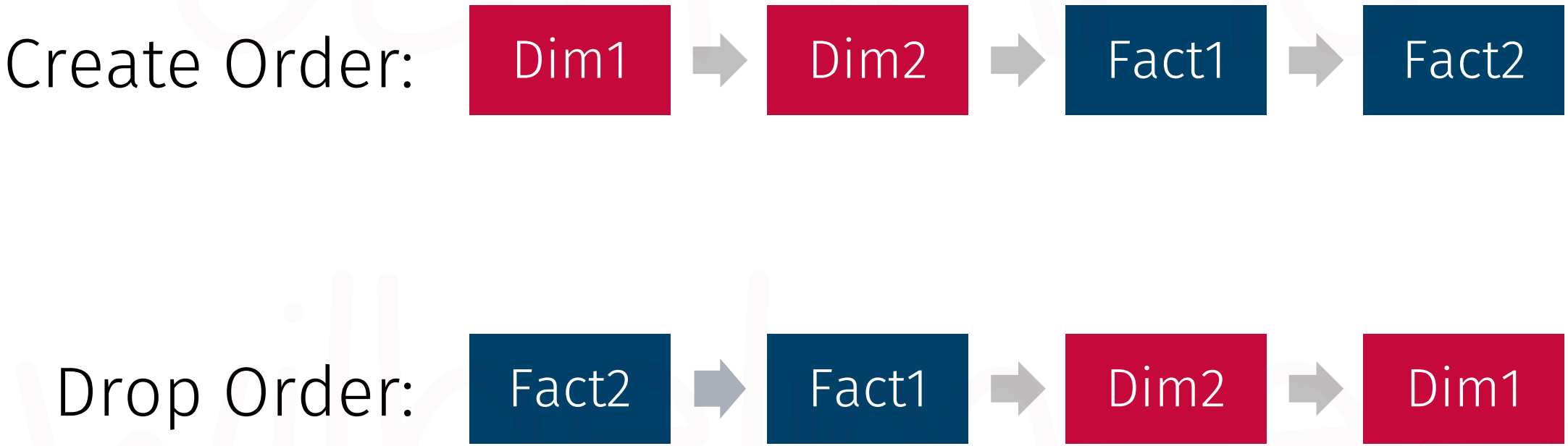
```
<# if (table.Sources.Any()) { #>

  <#=TableToPackageLowerer.GetStaticSourceInsertStatements(
    table.SchemaQualifiedName,
    table.HasIdentity,
    (AstTableStaticSourceNode)table.Sources.FirstOrDefault()
  )#>

<# } #>
```

Drop and Create

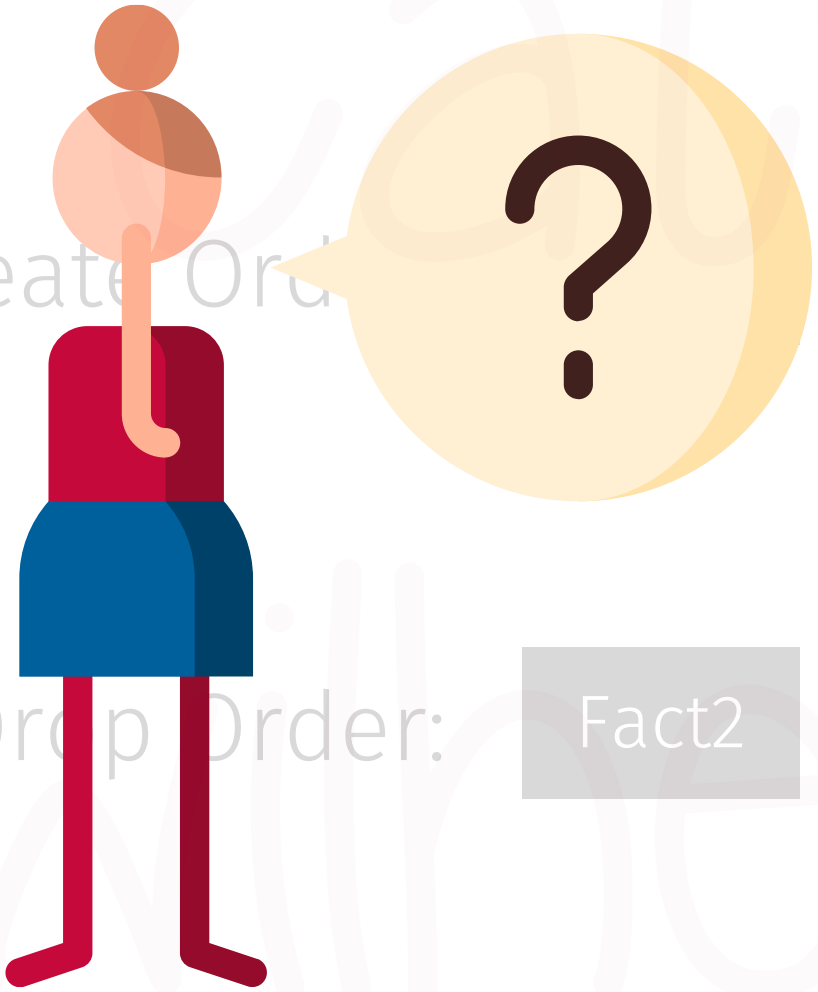
Facts and Dimensions

```
Could not drop object 'Schema.Table'
because it is referenced by a
FOREIGN KEY constraint.
```

Drop and Create



Drop and Create



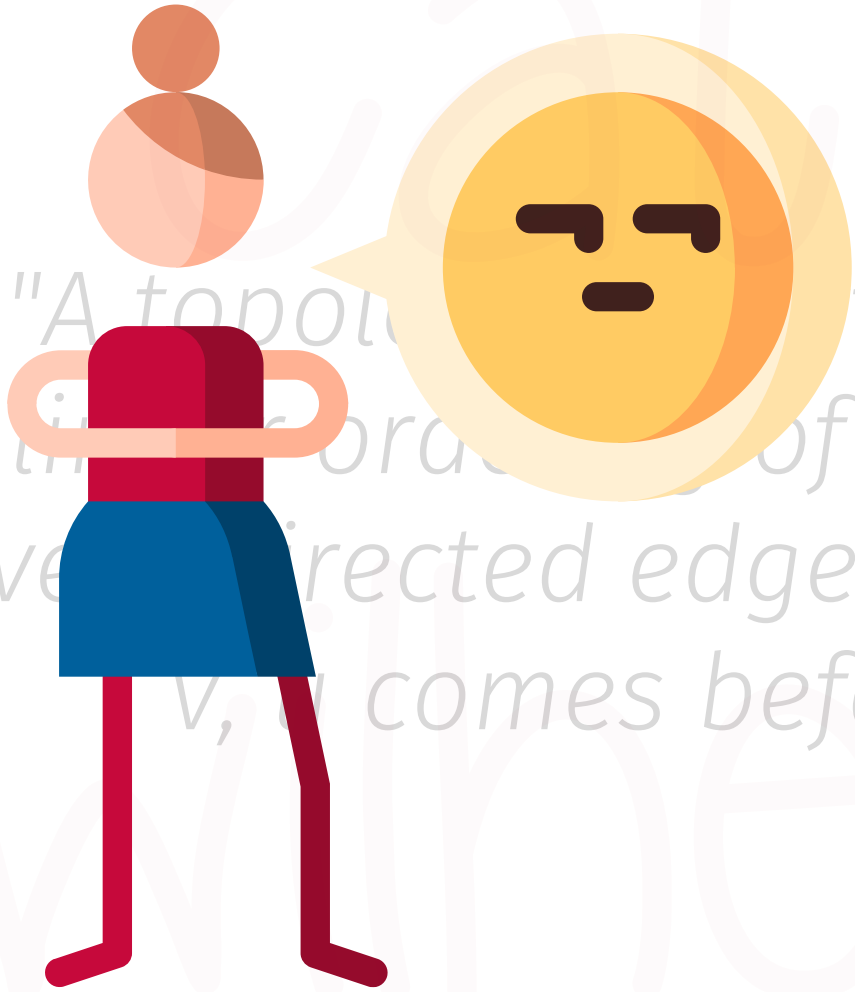
Create Order: ? → Dim2 → Fact1 → Fact2

Drop Order: Fact2 → Fact1 → Dim2 → Dim1

Topological Sort

"A topological sort of a directed graph is a linear ordering of its vertices such that for every directed edge uv from vertex u to vertex v, u comes before v in the ordering"

Topological Sort

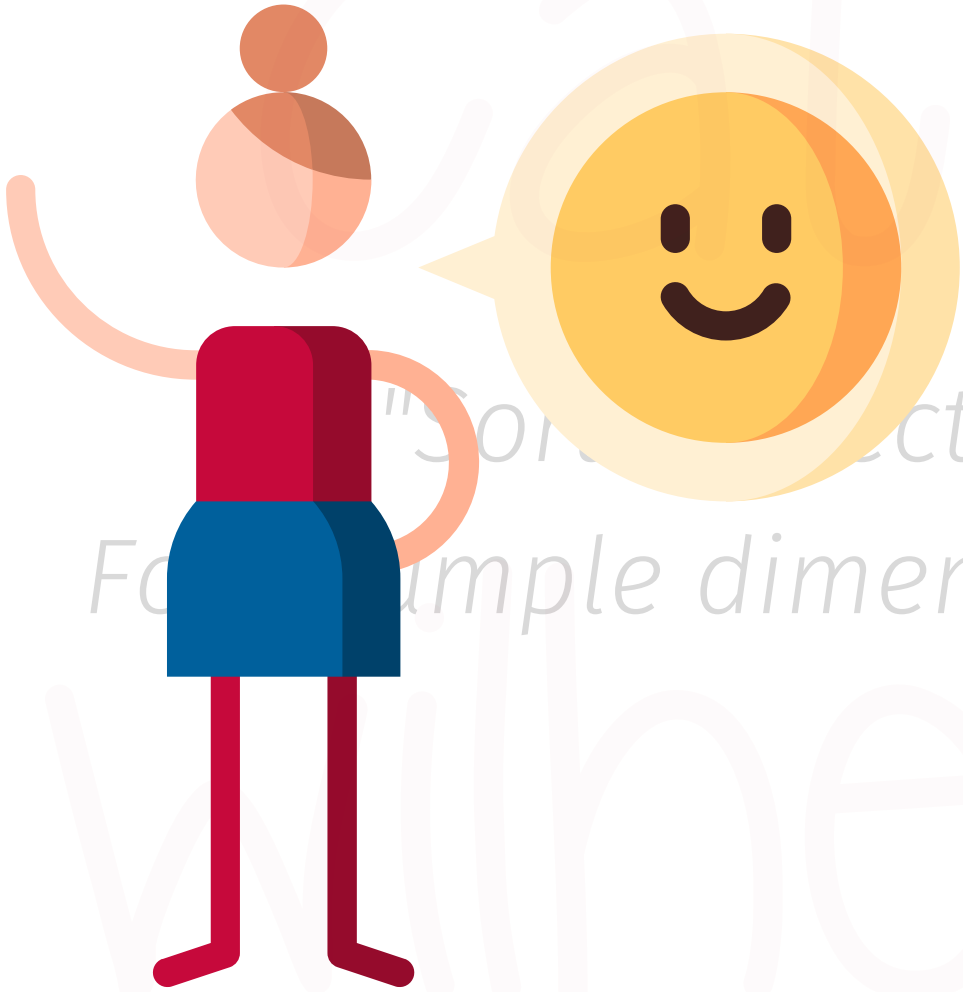


"A topological sort of a directed graph is a linear ordering of its vertices such that for every directed edge uv from vertex u to vertex v, u comes before v in the ordering"

Topological Sort

"Sorts objects by dependencies.
For example dimensions before fact tables."

Topological Sort



"Sorts objects by dependencies.
For example dimensions before fact tables."

Topological Sort

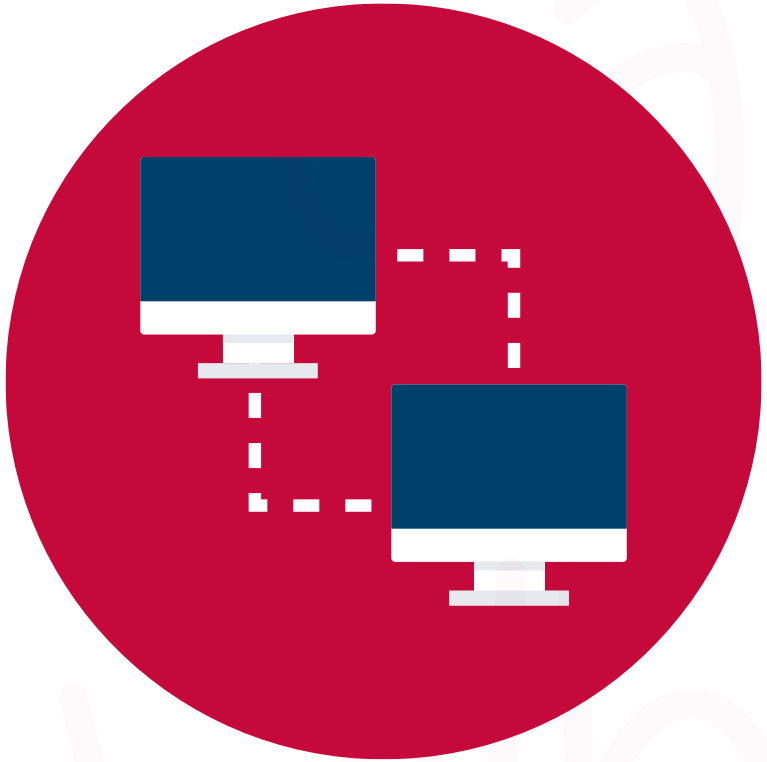
```
IEnumerable<T> DependencyAnalysis.TopologicalSort<T> (
      IEnumerable<T> source,
      Expression<Func<T, T>> relationProperty


)
```

Topological Sort

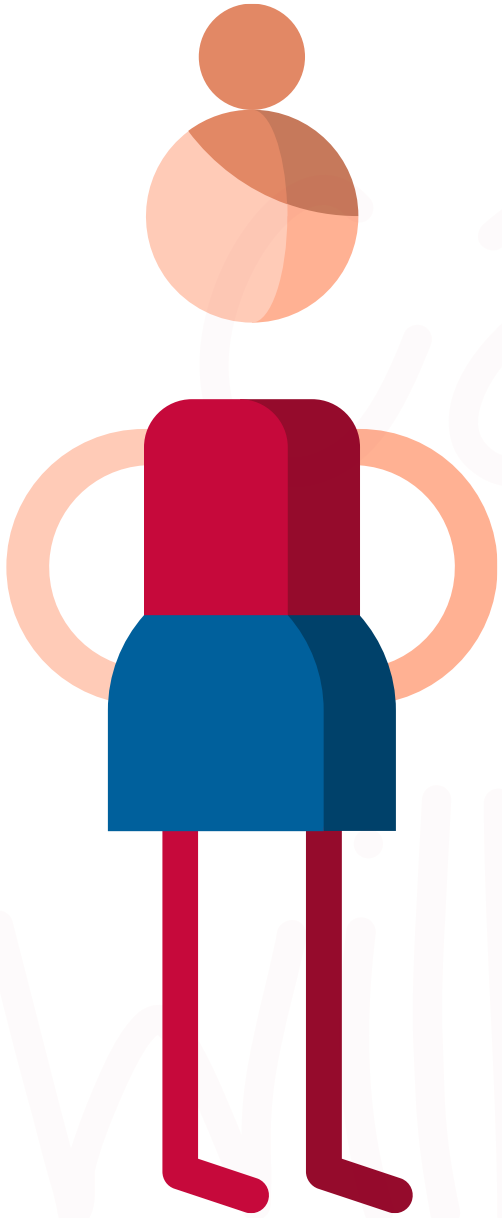
```
var dependentTables =
DependencyAnalysis.TopologicalSort<AstTableNode> (
      RootNode.Tables,
      t => t.Columns
           .OfType<AstTableColumnTableReferenceNode>()
           .Select(c => c.ForeignTable)
);
```

DEMO

Static Dimensions and Topological Sort

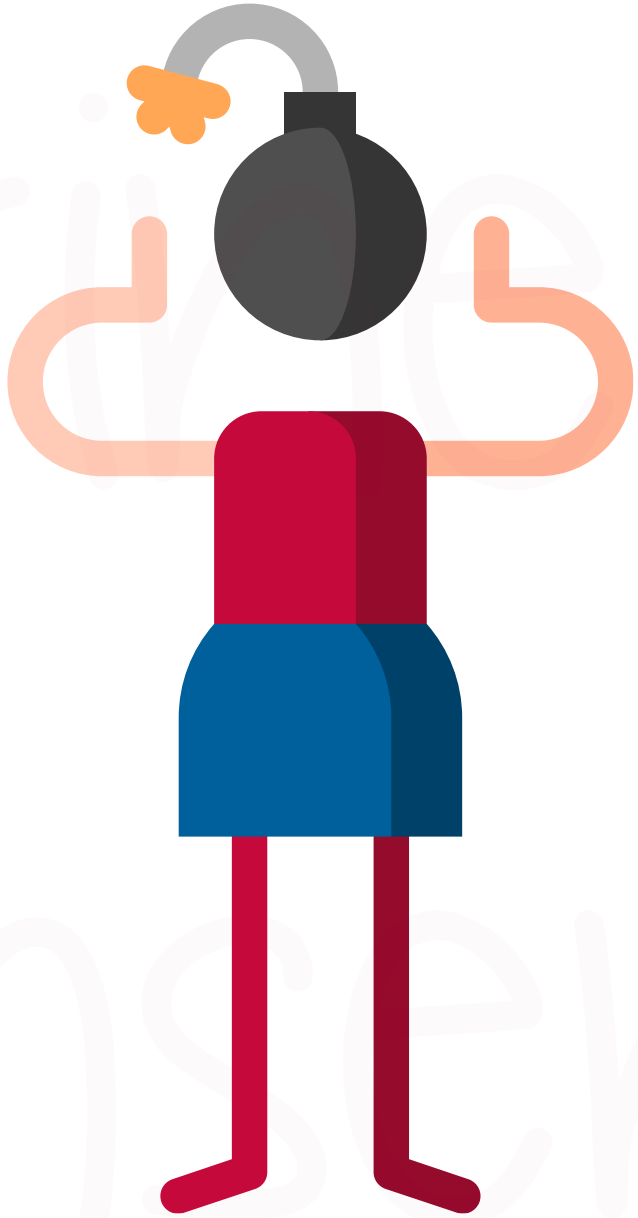


Comparing Databases



Once upon a time...

Panic!



How to Compare

Tables and Columns

LINQ

Filter
Where, OfType

Sort
OrderBy, ThenBy

Aggregate
Count, Sum

Group
GroupBy

Check Collections
All, Any, Contains

Set Operations
Except, Intersect

Project Collections
Select, SelectMany

LINQ

Filter

Where, OfType

Sort

OrderBy, ThenBy

Aggregate

Count, Sum

Group

GroupBy

Check Collections

All, Any, Contains

Set Operations

Except, **Intersect**

Project Collections

Select, SelectMany

LINQ: Filter Collections

`Where()`

Returns the filtered collection with all elements that meet the criteria

```
table.Where(t => t.Schema.Name == "dim")
```

LINQ: Set Operations

`Except()`

Returns elements in first collection that are *not* in second collection

```
table1.Except(table2)
```

LINQ: Set Operations

`Intersect()`

Returns elements in first collection that are *also* in second collection

```
table1.Intersect(table2)
```

LINQ: Project Collections

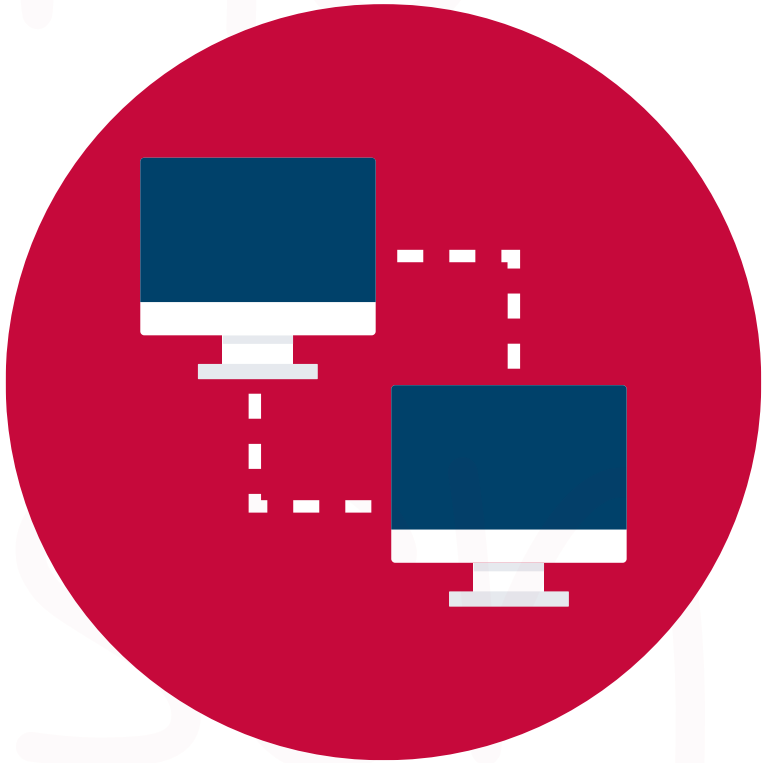
`Select()`

Creates a new collection from specified attributes

```
table.Select(t => t.Name)
```

DEMO

Comparing Databases



Alternative (*Better*) Solutions

Redgate SQL Compare (licensed)

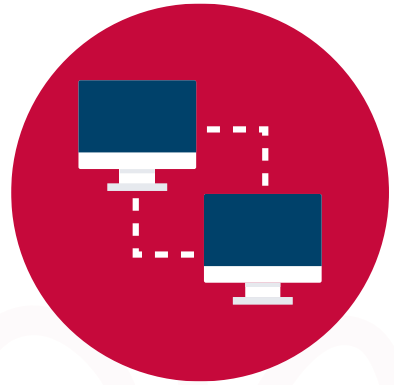
https://www.red-gate.com/products/sql-development/sql-compare/

...the past 60 minutes...

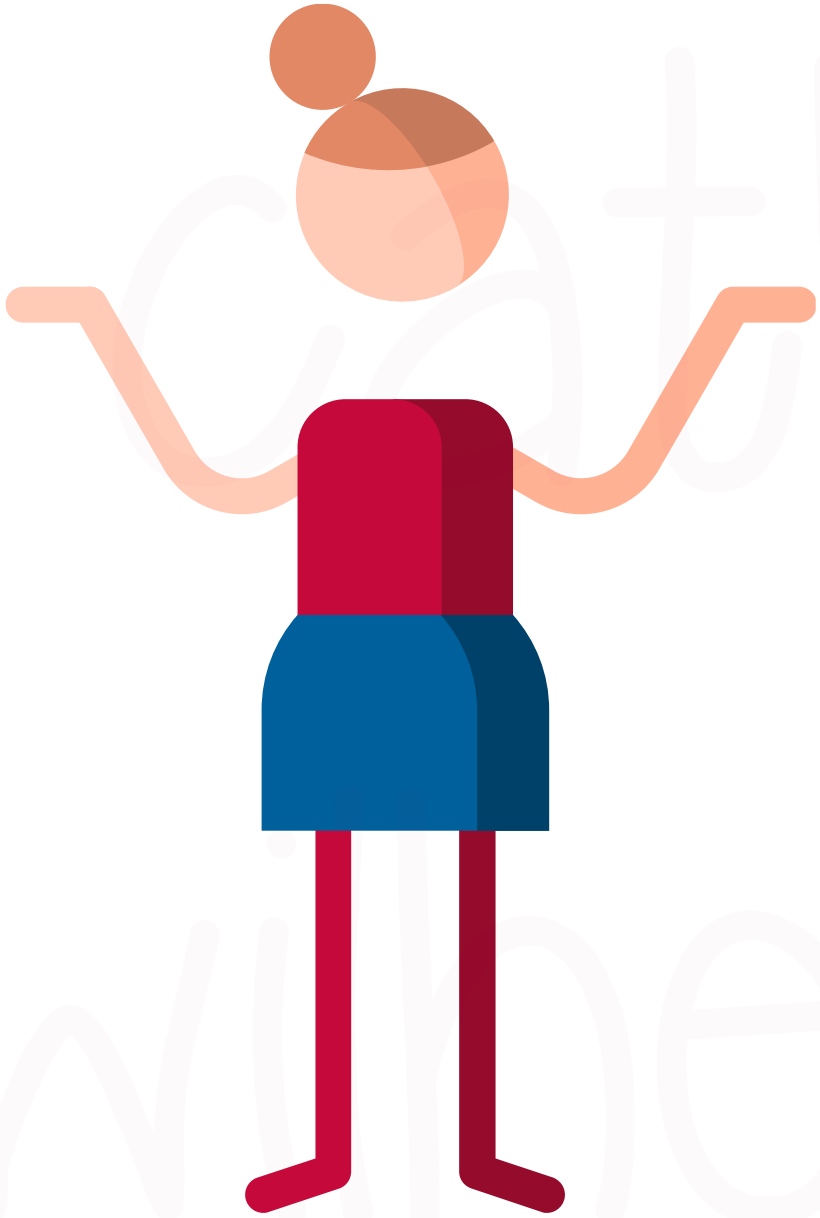
T-SQL

Test Data

Dimensions



Comparing



Is this useful?

What other ideas do you have?

Have fun!

Biml resources and demo files:

cathrinew.net/biml

hi@cathrinew.net

@cathrinew

CW

cathrinew.net