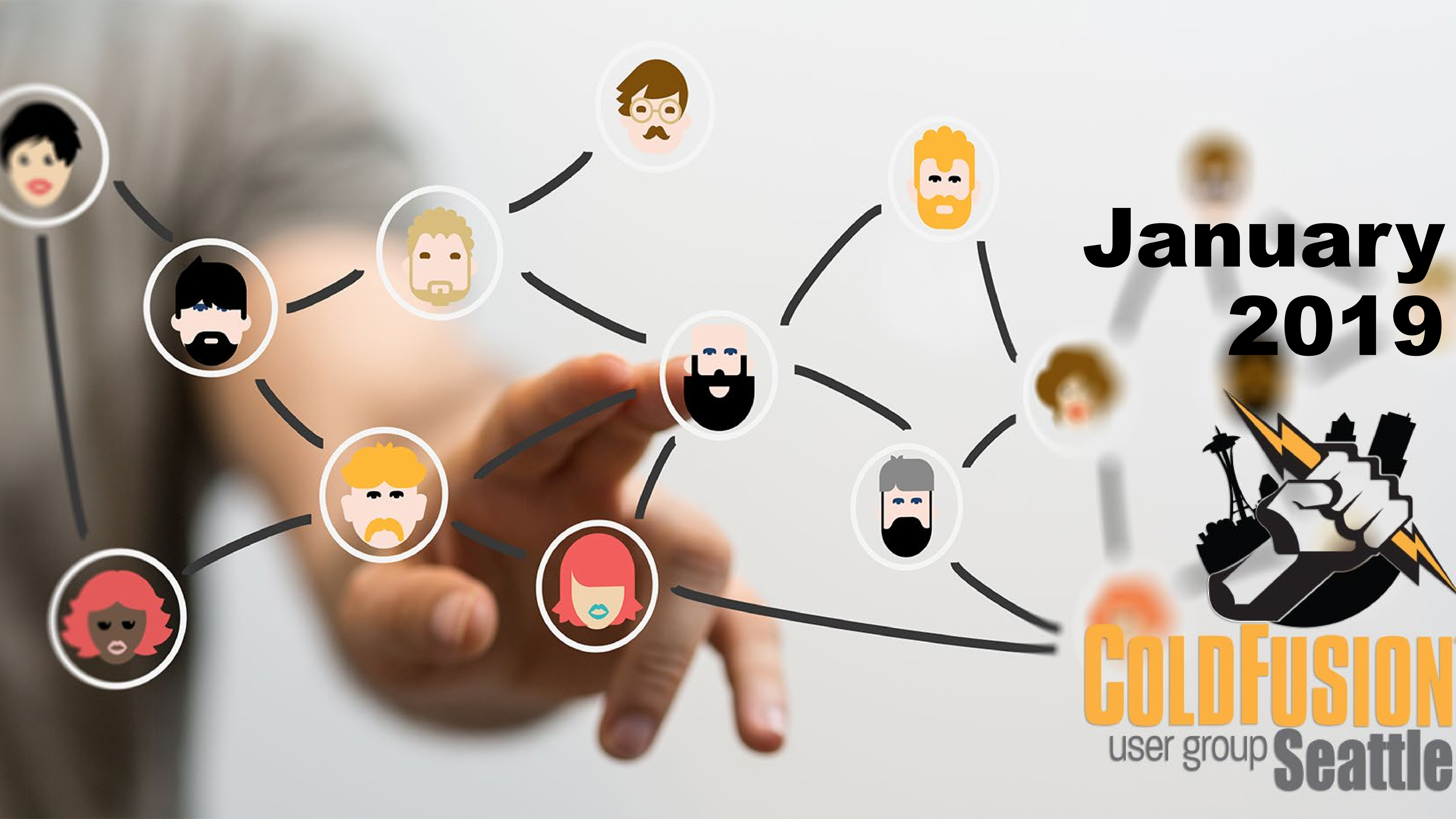


January 2019

ColdFusion user group Seattle

Agenda

- Welcome to the Seattle ColdFusion User Group
- Goals
- Introductions
- Leveraging SQL Server Stored Procedures and Scalar Functions in Your ColdFusion App
- The Importance of Proper Requirements Definition With Your Next ColdFusion Project
- CSS Trick for Alternating Data Grid Row Colors

Agenda

- February 2019 Seattle CFUG Meeting
- Questions/Answers/Help Needed

Goals

- Assist ColdFusion Developers Throughout the Pacific Northwest
- Promote ColdFusion Developers Throughout the Pacific Northwest
- Connect Employers with ColdFusion Developers
- Establish a Community of Friendship Between ColdFusion Developers
- Provide Speaking Opportunities for ColdFusion Developers
- Change the Perception of ColdFusion as a viable platform

Introductions

- Tell us a little bit about who you are
- Share with us what you would like to get from this user group

Leveraging Microsoft SQL Server Stored Procedures and Scalar Functions in your CF App

- What is a stored procedure and why should I use them?
- How do I use stored procedures in my ColdFusion app?
- What are scalar functions and why should I use them?
- How do I use scalar functions in my ColdFusion app?

Leveraging Microsoft SQL Server Stored Procedures and Scalar Functions in your CF App

- What is a stored procedure?
  A *stored procedure* is compiled tSQL code which can be efficiently reused

- Why should I use stored procedures?
  - Code Reuse (we love code reuse)
  - Obfuscate database objects
  - Dynamic SQL Reduction

Leveraging Microsoft SQL Server Stored Procedures and Scalar Functions in your CF App

- ColdFusion Tags for Stored Procedures
  - <cfprocedure> used to invoke a stored procedure https://cfdocs.org/cfstoredproc
  - <cfprocparam> used to pass parameters to/from stored procs https://cfdocs.org/cfprocparam
  - <cfprocresult> used to pass a recordset from a stored proc https://cfdocs.org/cfprocresult

Leveraging Microsoft SQL Server Stored Procedures and Scalar Functions in your CF App

<CFStoredProc> Example

```
<cfstoredproc datasource="#Application.RODSN#" procedure="getPointsEarnedByDateRangeAndLocation">
		<cfprocparam cfsqltype="CF_SQL_DATE" value="#VARIABLES.StartDate#">
		<cfprocparam cfsqltype="CF_SQL_DATE" value="#VARIABLES.EndDate#">
		<cfprocparam cfsqltype="CF_SQL_TINYINT" value="#VARIABLES.LocationId#">
		<cfprocresult resultset="1" name="VARIABLES.allData">
		<cfprocresult resultset="2" name="VARIABLES.groupedData">
</cfstoredproc>
```

Leveraging Microsoft SQL Server Stored Procedures and Scalar Functions in your CF App

- What are scalar functions?

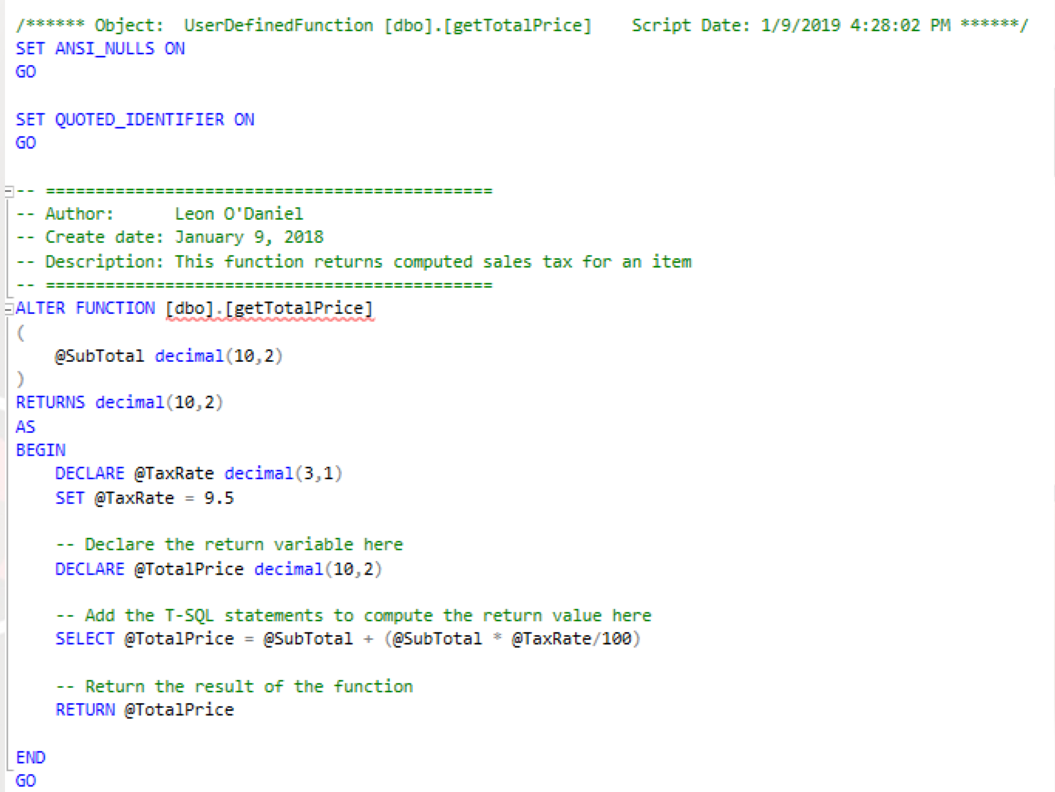
  Scalar functions are those that take one or more values…but only return a single value (and not a recordset). Many scalar functions are built-in to SQL Server (examples: SUM, Count, MIN, MAX, RTRIM, LTRIM)

  More info: https://www.w3schools.com/sql/sql_ref_sqlserver.asp and https://docs.microsoft.com/en-us/sql/t-sql/functions/functions?view=sql-server-2017

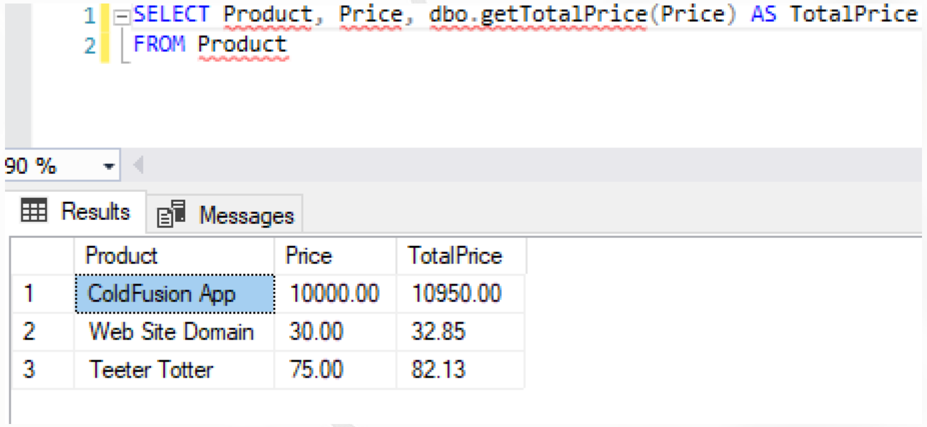
- When would I use scalar functions?
  - To simplify logic in a stored procedure?
  - To modify the value of a column returned from a query

Leveraging Microsoft SQL Server Stored Procedures and Scalar Functions in your CF App

- User Defined Scalar Functions



```
/****** Object:  UserDefinedFunction [dbo].[getTotalPrice]    Script Date: 1/9/2019 4:28:02 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:      Leon O'Daniel
-- Create date: January 9, 2018
-- Description: This function returns computed sales tax for an item
-- =============================================
ALTER FUNCTION [dbo].[getTotalPrice]
(
    @SubTotal decimal(10,2)
)
RETURNS decimal(10,2)
AS
BEGIN
    DECLARE @TaxRate decimal(3,1)
    SET @TaxRate = 9.5

    -- Declare the return variable here
    DECLARE @TotalPrice decimal(10,2)

    -- Add the T-SQL statements to compute the return value here
    SELECT @TotalPrice = @SubTotal + (@SubTotal * @TaxRate/100)

    -- Return the result of the function
    RETURN @TotalPrice

END
GO
```

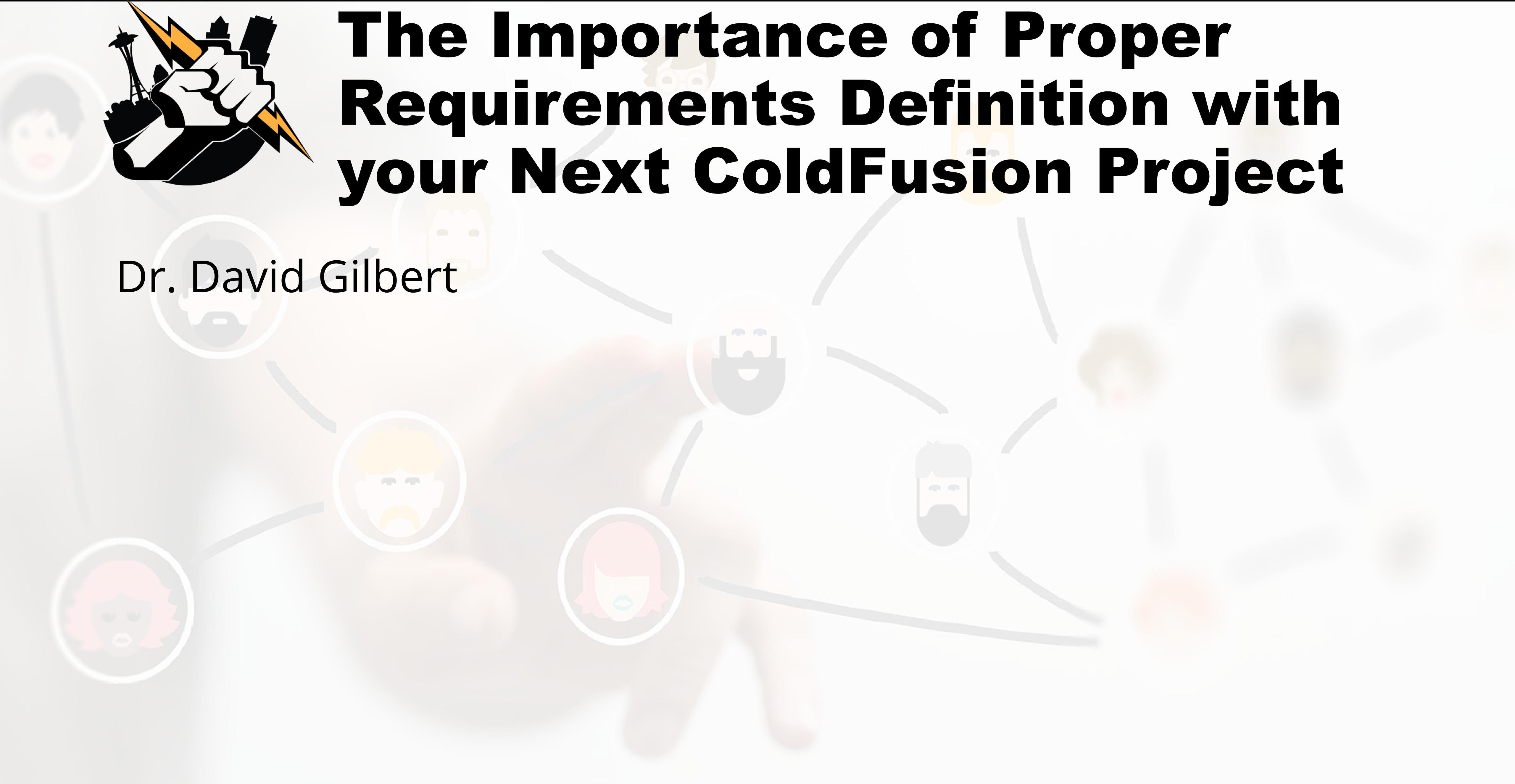


```
SELECT Product, Price, dbo.getTotalPrice(Price) AS TotalPrice
FROM Product
```

	Product	Price	TotalPrice
1	ColdFusion App	10000.00	10950.00
2	Web Site Domain	30.00	32.85
3	Teeter Totter	75.00	82.13

The Importance of Proper Requirements Definition with your Next ColdFusion Project

Dr. David Gilbert



CSS Trick for Alternating Data Grid Row Colors

- No longer need to use code like:

```
<cfif tablerecords.currentrow MOD(2)>
   <tr class=“gray”>
<cfelse>
   <tr class=“lightgray”>
</cfif>
```

CSS can handle this!

CSS Trick for Alternating Data Grid Row Colors

```
<style>
    tr:nth-child(odd) {background: #CCC}
    tr:nth-child(even) {background: #FFF}
</style>
```

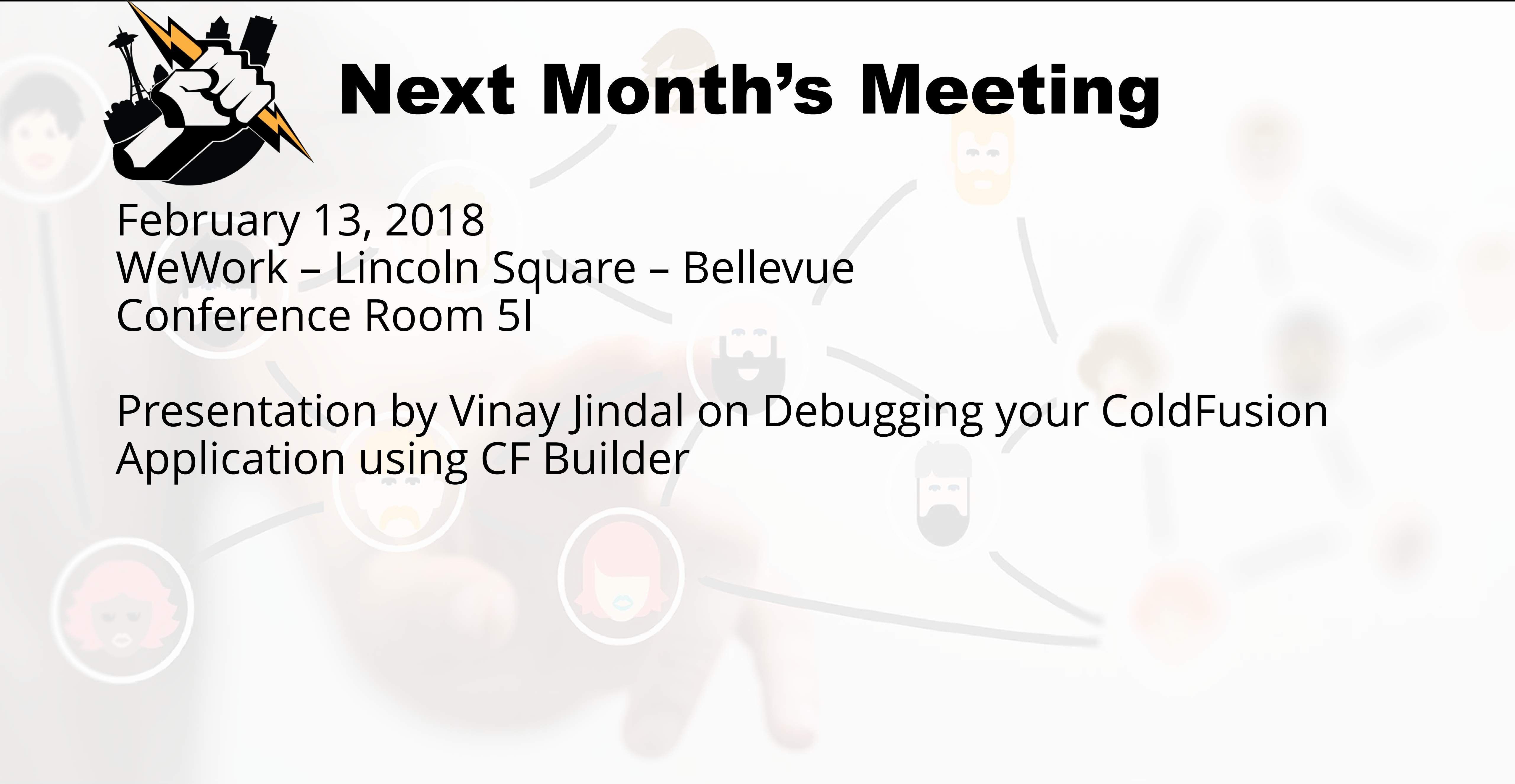
more info:

- https://caniuse.com/#feat=css-sel3
- https://www.w3.org/Style/Examples/007/evenodd.en.html

Next Month’s Meeting

February 13, 2018
WeWork – Lincoln Square – Bellevue
Conference Room 5I

Presentation by Vinay Jindal on Debugging your ColdFusion Application using CF Builder



Questions/Answers/Help!