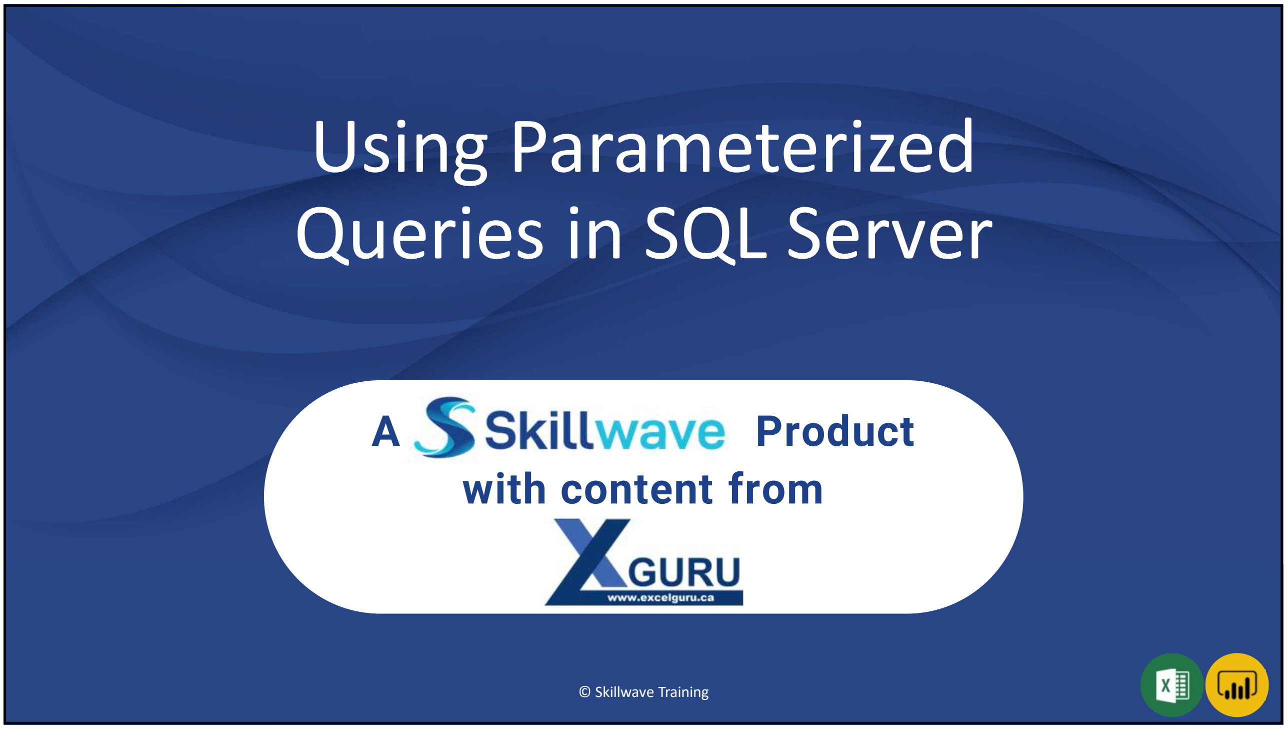
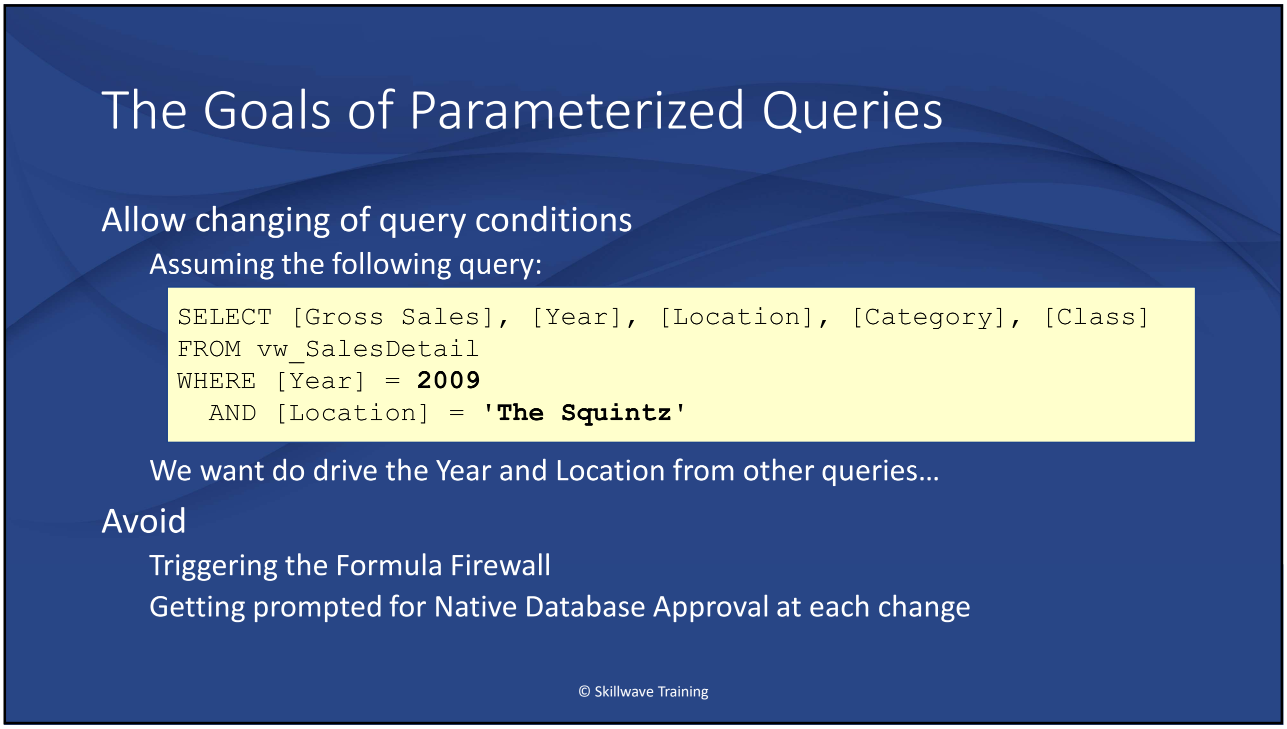


Using Parameterized Queries in SQL Server

A Skillwave Product
with content from

GURU
www.excelguru.ca



The Goals of Parameterized Queries

- Allow changing of query conditions
  - Assuming the following query:

```
SELECT [Gross Sales], [Year], [Location], [Category], [Class]
FROM vw_SalesDetail
WHERE [Year] = 2009
  AND [Location] = 'The Squintz'
```

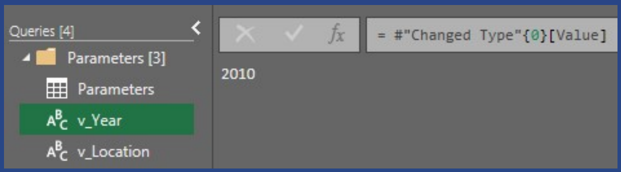
  - We want do drive the Year and Location from other queries…
- Avoid
  - Triggering the Formula Firewall
  - Getting prompted for Native Database Approval at each change

Step 1: Build “Scalar” Queries for all Variables

Steps:

- Connect to the data source
- Filter rows to your desired data point
- Remove all columns except the one that holds your data point
- Set the data type on the column
- Right click the data point value → Drill down
- Name the query
- Load as “Connection Only”

“Scalar” queries can be identified based on the fact that they have a data type icon, not a table icon.



`= #"Changed Type"{0}[Value]`

2010

Step 2: Create Static SQL Query & Result

Connect to the Database

- Right click the Database → Transform

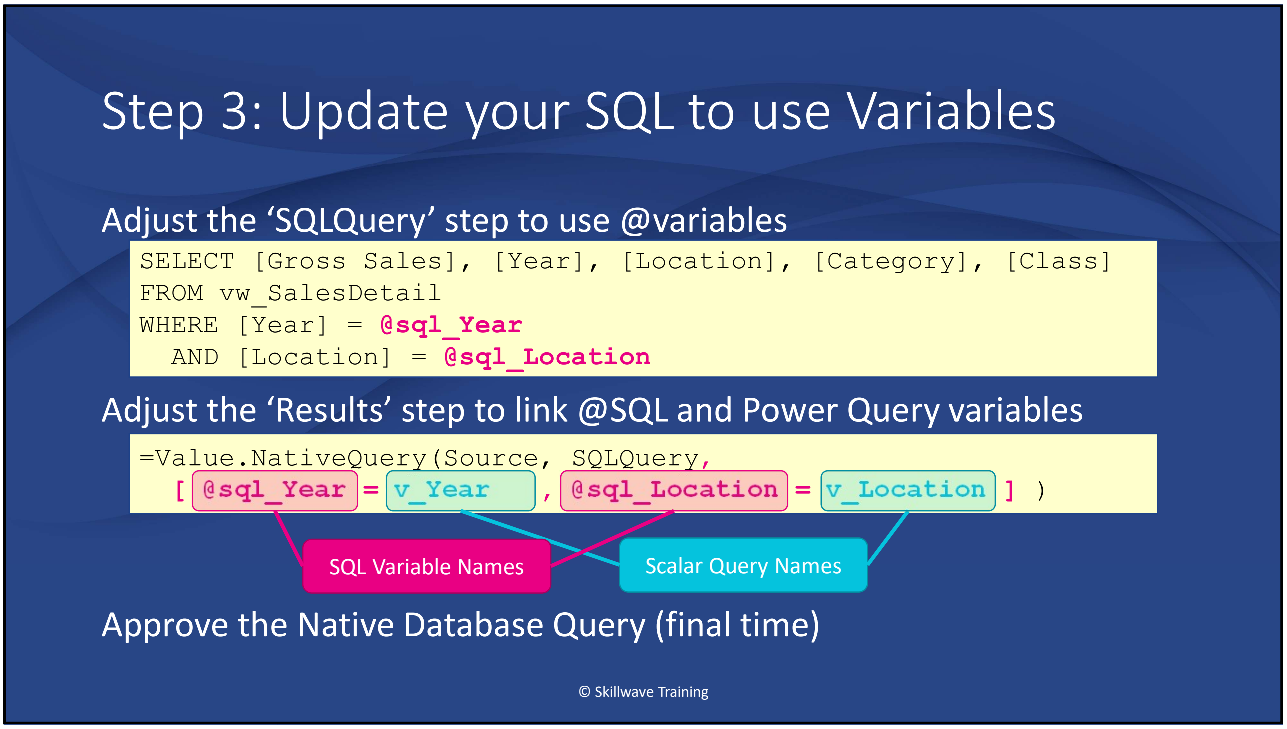
Create a “SQLQuery” step

- Click the fx button and rename the step
- Type in your SQL query

Create a “Results” step

- Click the fx button and rename the step
- Update the formula to: `=Value.NativeQuery(Source, SQLQuery)`
- Approve the Native Database Query permission

This step allows you to test that your SQL is correct, and that your database connection is working!



Step 3: Update your SQL to use Variables

Adjust the ‘SQLQuery’ step to use @variables

```
SELECT [Gross Sales], [Year], [Location], [Category], [Class]
FROM vw_SalesDetail
WHERE [Year] = @sql_Year
  AND [Location] = @sql_Location
```

Adjust the ‘Results’ step to link @SQL and Power Query variables

```
=Value.NativeQuery(Source, SQLQuery,
  [ @sql_Year = v_Year , @sql_Location = v_Location ] )
```

SQL Variable Names
Scalar Query Names

Approve the Native Database Query (final time)