Setting Up Queries in Stored Procedures

Jared Westover
SQL ARCHITECT

@WestoverJared

Module Overview

Adding queries to stored procedures

- Using temporary objects
- Updating values

Working with table-valued parameters

- Creating a table type
- Limitations

Executing code with sp_executesql

- Why use dynamic sql
- Comparing sp_executesql and exec

Troubleshooting made easy

- Debug flag

Purposes of Stored Procedures

Querying data

Return results back to the client

Modifying data

Perform updates insert and deletes

Temporary Objects

Reviewing temporary tables

- Why use temporary tables
- Creating and Dropping

Table variables

- Avoid for larger data sets
- Lack statistics

Common table expression

- Similar to a temp table

Disadvantages of temp objects

- They are temporary

```sql
DROP TABLE IF EXISTS #SalesOrderTemp;
GO
CREATE TABLE #SalesOrderTemp
    (SalesAmount decimal(36,2), Id int);
GO
```

Creating Temporary Tables

Try to match up the data type correctly to the underlining table

```
DECLARE @SalesOrderVar AS TABLE
   (SalesAmount decimal(36,2), Id int);
GO
```

Creating Table Variables

No need to drop since they are removed when batch is complete

Demo

Add an existing query to our sproc

- Insert sales person

Comparing temp tables and table variables

Table -valued Parameters

Why use table-valued parameters

- Passing several values

Creating a table type

- Define the data structure

Inserting data into our table type

- Correct data type

Cannot be modified

- Pass into a temporary table

```
CREATE TYPE SalesTableType
AS TABLE (SalesPersonEmail nvarchar(500));
GO
CREATE OR ALTER PROCEDURE Sales.SalesPersonReport
@SalesPersonEmail SalesTableType READONLY
```

Creating a Table Type

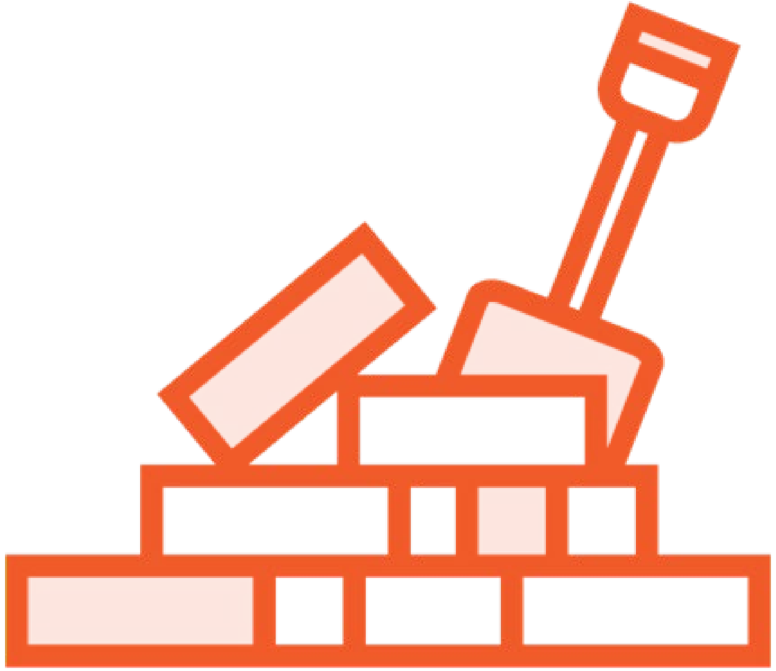
They cannot be altered once created

Demo

Create a table type and define the data structure

Pass values into a stored procedure with table-valued parameter

Examining Sp_executesql



What is sp_executesql

- Build sql statements dynamically

Why use dynamic sql

- Creating a dynamic query

Execution plan reuse

- Similar to our stored procedures

Compared to exec

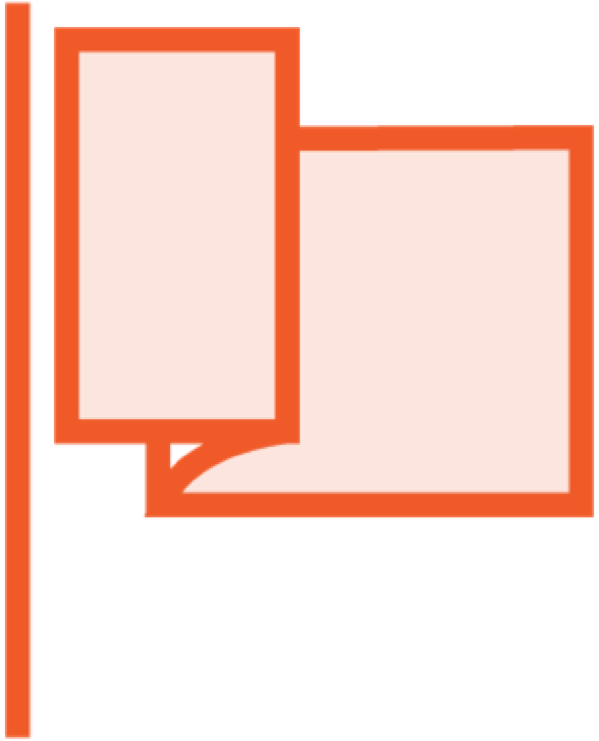
- Generally more secure
- Accepts parameters

Demo

Creating a dynamic query

- Execute with sp_executesql
- Execute with exec

Implementing a Debug Flag



Debugger can fall short

Intermediate result sets

- Temp tables

Sometimes called debug parameter

- Bit flag

Production usage

Setup time

- Worth it in the long run

Demo

Adding a debug flag after each statement

Executing our sproc with debug enabled

What We Covered

Added queries to our sprocs

- Temporary table
- Table variables

Table -valued parameters

- Pass several values
- Table type

Reviewed sp_executesql

- Dynamic sql
- Versus exec

Implemented a debug flag

- Review result sets

Next Module: Setting up Queries in Stored Procedures