Lab 04: Dynamic SQL

※\(^o^)/※

September, 2019

Contents

Introduction

This lab aims to help students get used to dynamic sql and related issues in Transact-SQL.

Lab Activities

EXEC Command

```sql
SET NOCOUNT ON;
USE SP;
GO

----------------------------------------
-- EXEC command
----------------------------------------


-- start snippet exec_example
```

```sql
DECLARE @s AS NVARCHAR(200);
SET @s = N'Davis'; -- originates in user input

DECLARE @sql AS NVARCHAR(1000);
SET @sql = N'SELECT empid, firstname, lastname, hiredate
FROM HR.Employees WHERE lastname = N''' + @s + N''';';

PRINT @sql; -- for debug purposes
EXEC (@sql);
-- end snippet exec_example
GO
```

SQL Injection

```sql
----------------------------------------
--SQL injection
----------------------------------------

-- start snippet sql_injection_01
DECLARE @s AS NVARCHAR(200);
SET @s = N'abc'' UNION ALL
SELECT object_id, SCHEMA_NAME(schema_id), name, NULL
FROM sys.objects WHERE type IN (''U'', ''V'');--';

DECLARE @sql AS NVARCHAR(1000);
SET @sql = N'SELECT empid, firstname, lastname, hiredate
FROM HR.Employees WHERE lastname = N''' + @s + N''';';

PRINT @sql; -- for debug purposes
EXEC (@sql);
-- end snippet sql_injection_01
GO

-- start snippet sql_injection_02
DECLARE @s AS NVARCHAR(200);
SET @s = N'abc'' UNION ALL
SELECT NULL, name, NULL, NULL
FROM sys.columns WHERE object_id = 1141579105; --';

DECLARE @sql AS NVARCHAR(1000);
SET @sql = N'SELECT empid, firstname, lastname, hiredate
FROM HR.Employees WHERE lastname = N''' + @s + N''';';

PRINT @sql; -- for debug purposes
EXEC (@sql);
```

```sql
-- end snippet sql_injection_02
GO

-- start snippet sql_injection_03
DECLARE @s AS NVARCHAR(200);
SET @s = N'abc'' UNION ALL
SELECT NULL, companyname, phone, NULL
FROM Sales.Customers; --';

DECLARE @sql AS NVARCHAR(1000);
SET @sql = N'SELECT empid, firstname, lastname, hiredate
FROM HR.Employees WHERE lastname = N''' + @s + N''';';

PRINT @sql; -- for debug purposes
EXEC (@sql);
-- end snippet sql_injection_03
GO
```

sp_executesql Procedure

```sql
----------------------------------------
-- sp_executesql
----------------------------------------

-- Input Parameters
-- start snippet sp_executesql_example
DECLARE @s AS NVARCHAR(200);
SET @s = N'Davis';

DECLARE @sql AS NVARCHAR(1000);
SET @sql = 'SELECT empid, firstname, lastname, hiredate
FROM HR.Employees WHERE lastname = @lastname;';

PRINT @sql; -- For debug purposes

EXEC sp_executesql
  @stmt = @sql,
  @params = N'@lastname AS NVARCHAR(200)',
  @lastname = @s;
-- end snippet sp_executesql_example
GO
```

Dynamic Search Conditions

```
----------------------------------------
-- Dynamic search conditions
----------------------------------------

-- Create tables and indices
SET NOCOUNT ON;
USE tempdb;
GO

IF OBJECT_ID(N'dbo.Orders', N'U') IS NOT NULL DROP TABLE dbo.Orders;
GO

SELECT orderid, custid, empid, orderdate,
  CAST('A' AS CHAR(200)) AS filler
INTO dbo.Orders
FROM SP.Sales.Orders;

CREATE CLUSTERED INDEX idx_orderdate ON dbo.Orders(orderdate);
CREATE UNIQUE INDEX idx_orderid ON dbo.Orders(orderid);
CREATE INDEX idx_custid_empid ON dbo.Orders(custid, empid)
        INCLUDE(orderid, orderdate, filler);
GO

-- Create dbo.getOrders stored procedure
-- with optional inputs:
-- @orderid, @custid, @empid, and @orderdate
-- query and filter on non-NULL input values

-- Use static query
USE tempdb;
IF OBJECT_ID(N'dbo.GetOrders', N'P') IS NOT NULL DROP PROC dbo.GetOrders;
GO
--start snippet static_ds
CREATE PROC dbo.GetOrders
  @orderid   AS INT  = NULL,
  @custid    AS INT  = NULL,
  @empid     AS INT  = NULL,
  @orderdate AS DATE = NULL
AS

SELECT orderid, custid, empid, orderdate, filler
FROM dbo.Orders
WHERE (orderid   = @orderid   OR @orderid   IS NULL)
```

```
  AND (custid    = @custid    OR @custid    IS NULL)
  AND (empid     = @empid     OR @empid     IS NULL)
  AND (orderdate = @orderdate OR @orderdate IS NULL);
--end snippet static_ds
GO

-- Test procedure
USE tempdb;
--start snippet static_ds_test
EXEC dbo.GetOrders @orderdate = '20140101';
--end snippet static_ds_test
GO

--Use static query with OPTION(RECOMPILE)
--at statement level
USE tempdb;
IF OBJECT_ID(N'dbo.GetOrders', N'P') IS NOT NULL DROP PROC dbo.GetOrders;
GO
--start snippet static_ds_rc
CREATE PROC dbo.GetOrders
  @orderid   AS INT  = NULL,
  @custid    AS INT  = NULL,
  @empid     AS INT  = NULL,
  @orderdate AS DATE = NULL
AS
SELECT orderid, custid, empid, orderdate, filler
FROM dbo.Orders
WHERE (orderid   = @orderid   OR @orderid   IS NULL)
  AND (custid    = @custid    OR @custid    IS NULL)
  AND (empid     = @empid     OR @empid     IS NULL)
  AND (orderdate = @orderdate OR @orderdate IS NULL)
OPTION (RECOMPILE);
--end snippet static_ds_rc
GO

-- Test procedure
USE tempdb;
--start snippet static_ds_rc_test
EXEC dbo.GetOrders @orderdate = '20140101';
EXEC dbo.GetOrders @orderid   = 10248;
--end snippet static_ds_rc_test
GO

-- Use dynamic SQL with parameters.
USE tempdb;
IF OBJECT_ID(N'dbo.GetOrders', N'P') IS NOT NULL DROP PROC dbo.GetOrders;
```

```
GO
--start snippet dynamic_ds_01
CREATE PROC dbo.GetOrders
  @orderid   AS INT  = NULL,
  @custid    AS INT  = NULL,
  @empid     AS INT  = NULL,
  @orderdate AS DATE = NULL
AS
DECLARE @sql AS NVARCHAR(1000);
SET @sql =
    N'SELECT orderid, custid, empid, orderdate, filler'
  + N' /* 27702431-107C-478C-8157-6DFCECC148DD */'
  + N' FROM dbo.Orders'
  + N' WHERE 1 = 1'
  + CASE WHEN @orderid IS NOT NULL THEN
      N' AND orderid = @oid' ELSE N'' END
--end snippet dynamic_ds_01
--start snippet dynamic_ds_02
  + CASE WHEN @custid IS NOT NULL THEN
      N' AND custid = @cid' ELSE N'' END
  + CASE WHEN @empid IS NOT NULL THEN
      N' AND empid = @eid' ELSE N'' END
  + CASE WHEN @orderdate IS NOT NULL THEN
      N' AND orderdate = @dt' ELSE N'' END;
EXEC sp_executesql
  @stmt = @sql,
  @params = N'@oid AS INT, @cid AS INT, @eid AS INT, @dt AS DATE',
  @oid = @orderid,
  @cid = @custid,
  @eid = @empid,
  @dt  = @orderdate;
--end snippet dynamic_ds_02
GO

-- Test procedure
USE tempdb;
--start snippet dynamic_ds_test
EXEC dbo.GetOrders @orderdate = '20140101';
EXEC dbo.GetOrders @orderdate = '20140102';
EXEC dbo.GetOrders @orderid   = 10248;
--end snippet dynamic_ds_test
GO


-- Find cached plans and their reuse
USE tempdb;
```

```sql
--start snippet cached_plans
SELECT usecounts, text
FROM sys.dm_exec_cached_plans AS CP
  CROSS APPLY sys.dm_exec_sql_text(cp.plan_handle) AS ST
WHERE ST.text LIKE '%27702431-107C-478C-8157-6DFCECC148DD%'
  AND ST.text NOT LIKE '%sys.dm_exec_cached_plans%'
  AND CP.objtype = 'Prepared';
--end snippet cached_plans
GO
```

Dynamic Sorting

```sql
---------------------------------------------
--Dynamic Sorting
---------------------------------------------

--Use static SQL with recompile
USE SP;
IF OBJECT_ID(N'dbo.GetSortedShippers', N'P') IS NOT NULL DROP PROC dbo.GetSortedShippers;
GO
--start snippet static_dsrt_rc
CREATE PROC dbo.GetSortedShippers
  @colname AS sysname, @sortdir AS CHAR(1) = 'A'
AS

SELECT shipperid, companyname, phone
FROM Sales.Shippers
--end snippet static_dsrt_rc
--start snippet static_dsrt_rc_02
ORDER BY
  CASE WHEN @colname = N'shipperid'   AND @sortdir = 'A'
        THEN shipperid   END,
  CASE WHEN @colname = N'companyname' AND @sortdir = 'A'
        THEN companyname END,
  CASE WHEN @colname = N'phone'       AND @sortdir = 'A'
        THEN phone       END,
  CASE WHEN @colname = N'shipperid'   AND @sortdir = 'D'
        THEN shipperid   END DESC,
  CASE WHEN @colname = N'companyname' AND @sortdir = 'D'
        THEN companyname END DESC,
  CASE WHEN @colname = N'phone'       AND @sortdir = 'D'
        THEN phone       END DESC
OPTION (RECOMPILE);
--end snippet static_dsrt_rc_02
GO
```

```

-- Test procedure

--start snippet static_dsrt_rc_test
EXEC dbo.GetSortedShippers N'shipperid', N'D';
--end snippet static_dsrt_rc_test
GO

-- Use dynamic SQL
USE SP;
IF OBJECT_ID(N'dbo.GetSortedShippers', N'P') IS NOT NULL DROP PROC dbo.GetSortedShippers;
GO

--start snippet dynamic_dsrt
CREATE PROC dbo.GetSortedShippers
  @colname AS sysname, @sortdir AS CHAR(1) = 'A'
AS
IF @colname NOT IN(N'shipperid', N'companyname', N'phone')
  THROW 50001,
        'Column name not supported. Possibly a SQL injection attempt.', 1;

DECLARE @sql AS NVARCHAR(1000);
SET @sql = N'SELECT shipperid, companyname, phone
FROM Sales.Shippers
ORDER BY '
  + QUOTENAME(@colname) +
        CASE @sortdir WHEN 'D' THEN N' DESC' ELSE '' END + ';';

EXEC sys.sp_executesql @stmt = @sql;
--end snippet dynamic_dsrt
GO

-- Test procedure
--start snippet dynamic_dsrt_test
EXEC dbo.GetSortedShippers N'shipperid', N'D';
--end snippet dynamic_dsrt_test
GO

-- Sort by multiple columns
USE SP;
IF OBJECT_ID(N'dbo.GetSortedShippers', N'P') IS NOT NULL DROP PROC dbo.GetSortedShippers;
GO

--start snippet dynamic_dsrt_mc_01
CREATE PROC dbo.GetSortedShippers
  @colname1 AS sysname, @sortdir1 AS CHAR(1) = 'A',
  @colname2 AS sysname = NULL, @sortdir2 AS CHAR(1) = 'A',
  @colname3 AS sysname = NULL, @sortdir3 AS CHAR(1) = 'A'
AS
IF @colname1 NOT IN(N'shipperid', N'companyname', N'phone')
   OR @colname2 IS NOT NULL
        AND @colname2 NOT IN(N'shipperid', N'companyname', N'phone')
   OR @colname3 IS NOT NULL
        AND @colname3 NOT IN(N'shipperid', N'companyname', N'phone')
  THROW 50001,
        'Column name not supported. Possibly a SQL injection attempt.', 1;
--end snippet dynamic_dsrt_mc_01

--start snippet dynamic_dsrt_mc_02
DECLARE @sql AS NVARCHAR(1000);

SET @sql = N'SELECT shipperid, companyname, phone
FROM Sales.Shippers
ORDER BY '
  + QUOTENAME(@colname1) +
        CASE @sortdir1 WHEN 'D' THEN N' DESC' ELSE '' END
  + ISNULL(N',' + QUOTENAME(@colname2) +
        CASE @sortdir2 WHEN 'D' THEN N' DESC' ELSE '' END, N'')
  + ISNULL(N',' + QUOTENAME(@colname3) +
        CASE @sortdir3 WHEN 'D' THEN N' DESC' ELSE '' END, N'')
  + ';';

EXEC sys.sp_executesql @stmt = @sql;
--end snippet dynamic_dsrt_mc_02
GO
```