

# Lab 04: Dynamic SQL

※\(\^o\^)/※

September, 2019

## Contents

<b>Introduction</b>	<b>1</b>
<b>Lab Activities</b>	<b>1</b>
EXEC Command . . . . .	1
SQL Injection . . . . .	2
sp_executesql Procedure . . . . .	3
Dynamic Search Conditions . . . . .	4
Dynamic Sorting . . . . .	7

## Introduction

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

## Lab Activities

### EXEC Command

```
5 SET NOCOUNT ON;
6 USE SP;
7 GO
8
9 -----
10 -- EXEC command
11 -----
12
13
14 -- start snippet exec_example
```

```

15 DECLARE @s AS NVARCHAR(200);
16 SET @s = N'Davis'; -- originates in user input
17
18 DECLARE @sql AS NVARCHAR(1000);
19 SET @sql = N'SELECT empid, firstname, lastname, hiredate
20 FROM HR.Employees WHERE lastname = N''' + @s + N''';';
21
22 PRINT @sql; -- for debug purposes
23 EXEC (@sql);
24 -- end snippet exec_example
25 GO

```

## SQL Injection

```

27 -----
28 --SQL injection
29 -----
30
31 -- start snippet sql_injection_01
32 DECLARE @s AS NVARCHAR(200);
33 SET @s = N'abc'' UNION ALL
34 SELECT object_id, SCHEMA_NAME(schema_id), name, NULL
35 FROM sys.objects WHERE type IN (''U'', ''V'');--';
36
37 DECLARE @sql AS NVARCHAR(1000);
38 SET @sql = N'SELECT empid, firstname, lastname, hiredate
39 FROM HR.Employees WHERE lastname = N''' + @s + N''';';
40
41 PRINT @sql; -- for debug purposes
42 EXEC (@sql);
43 -- end snippet sql_injection_01
44 GO
45
46 -- start snippet sql_injection_02
47 DECLARE @s AS NVARCHAR(200);
48 SET @s = N'abc'' UNION ALL
49 SELECT NULL, name, NULL, NULL
50 FROM sys.columns WHERE object_id = 1141579105; --';
51
52 DECLARE @sql AS NVARCHAR(1000);
53 SET @sql = N'SELECT empid, firstname, lastname, hiredate
54 FROM HR.Employees WHERE lastname = N''' + @s + N''';';
55
56 PRINT @sql; -- for debug purposes
57 EXEC (@sql);

```

```

58  -- end snippet sql_injection_02
59  GO
60
61  -- start snippet sql_injection_03
62  DECLARE @s AS NVARCHAR(200);
63  SET @s = N'abc'' UNION ALL
64  SELECT NULL, companyname, phone, NULL
65  FROM Sales.Customers; --';
66
67  DECLARE @sql AS NVARCHAR(1000);
68  SET @sql = N'SELECT empid, firstname, lastname, hiredate
69  FROM HR.Employees WHERE lastname = N'' + @s + N'';';
70
71  PRINT @sql; -- for debug purposes
72  EXEC (@sql);
73  -- end snippet sql_injection_03
74  GO

```

## sp\_executesql Procedure

```

77  -----
78  -- sp_executesql
79  -----
80
81  -- Input Parameters
82  -- start snippet sp_executesql_example
83  DECLARE @s AS NVARCHAR(200);
84  SET @s = N'Davis';
85
86  DECLARE @sql AS NVARCHAR(1000);
87  SET @sql = 'SELECT empid, firstname, lastname, hiredate
88  FROM HR.Employees WHERE lastname = @lastname;';
89
90  PRINT @sql; -- For debug purposes
91
92  EXEC sp_executesql
93      @stmt = @sql,
94      @params = N'@lastname AS NVARCHAR(200)',
95      @lastname = @s;
96  -- end snippet sp_executesql_example
97  GO

```

## Dynamic Search Conditions

```
99  -----
100  -- Dynamic search conditions
101  -----
102
103  -- Create tables and indices
104  SET NOCOUNT ON;
105  USE tempdb;
106  GO
107
108  IF OBJECT_ID(N'dbo.Orders', N'U') IS NOT NULL DROP TABLE dbo.Orders;
109  GO
110
111  SELECT orderid, custid, empid, orderdate,
112         CAST('A' AS CHAR(200)) AS filler
113  INTO dbo.Orders
114  FROM SP.Sales.Orders;
115
116  CREATE CLUSTERED INDEX idx_orderdate ON dbo.Orders(orderdate);
117  CREATE UNIQUE INDEX idx_orderid ON dbo.Orders(orderid);
118  CREATE INDEX idx_custid_empid ON dbo.Orders(custid, empid)
119         INCLUDE(orderid, orderdate, filler);
120  GO
121
122  -- Create dbo.getOrders stored procedure
123  -- with optional inputs:
124  -- @orderid, @custid, @empid, and @orderdate
125  -- query and filter on non-NULL input values
126
127  -- Use static query
128  USE tempdb;
129  IF OBJECT_ID(N'dbo.GetOrders', N'P') IS NOT NULL DROP PROC dbo.GetOrders;
130  GO
131  --start snippet static_ds
132  CREATE PROC dbo.GetOrders
133  @orderid AS INT = NULL,
134  @custid AS INT = NULL,
135  @empid AS INT = NULL,
136  @orderdate AS DATE = NULL
137  AS
138
139  SELECT orderid, custid, empid, orderdate, filler
140  FROM dbo.Orders
141  WHERE (orderid = @orderid OR @orderid IS NULL)
```

```

142     AND (custid      = @custid      OR @custid      IS NULL)
143     AND (empid       = @empid       OR @empid       IS NULL)
144     AND (orderdate = @orderdate OR @orderdate IS NULL);
145 --end snippet static_ds
146 GO
147
148 -- Test procedure
149 USE tempdb;
150 --start snippet static_ds_test
151 EXEC dbo.GetOrders @orderdate = '20140101';
152 --end snippet static_ds_test
153 GO
154
155 --Use static query with OPTION(RECOMPILE)
156 --at statement level
157 USE tempdb;
158 IF OBJECT_ID(N'dbo.GetOrders', N'P') IS NOT NULL DROP PROC dbo.GetOrders;
159 GO
160 --start snippet static_ds_rc
161 CREATE PROC dbo.GetOrders
162     @orderid  AS INT = NULL,
163     @custid   AS INT = NULL,
164     @empid    AS INT = NULL,
165     @orderdate AS DATE = NULL
166 AS
167     SELECT orderid, custid, empid, orderdate, filler
168 FROM dbo.Orders
169 WHERE (orderid  = @orderid  OR @orderid  IS NULL)
170     AND (custid   = @custid   OR @custid   IS NULL)
171     AND (empid    = @empid    OR @empid    IS NULL)
172     AND (orderdate = @orderdate OR @orderdate IS NULL)
173 OPTION (RECOMPILE);
174 --end snippet static_ds_rc
175 GO
176
177 -- Test procedure
178 USE tempdb;
179 --start snippet static_ds_rc_test
180 EXEC dbo.GetOrders @orderdate = '20140101';
181 EXEC dbo.GetOrders @orderid   = 10248;
182 --end snippet static_ds_rc_test
183 GO
184
185 -- Use dynamic SQL with parameters.
186 USE tempdb;
187 IF OBJECT_ID(N'dbo.GetOrders', N'P') IS NOT NULL DROP PROC dbo.GetOrders;

```

```

188 GO
189 --start snippet dynamic_ds_01
190 CREATE PROC dbo.GetOrders
191     @orderid AS INT = NULL,
192     @custid AS INT = NULL,
193     @empid AS INT = NULL,
194     @orderdate AS DATE = NULL
195 AS
196 DECLARE @sql AS NVARCHAR(1000);
197 SET @sql =
198     N'SELECT orderid, custid, empid, orderdate, filler'
199     + N' /* 27702431-107C-478C-8157-6DFCECC148DD */'
200     + N' FROM dbo.Orders'
201     + N' WHERE 1 = 1'
202     + CASE WHEN @orderid IS NOT NULL THEN
203         N' AND orderid = @oid' ELSE N'' END
204 --end snippet dynamic_ds_01
205 --start snippet dynamic_ds_02
206     + CASE WHEN @custid IS NOT NULL THEN
207         N' AND custid = @cid' ELSE N'' END
208     + CASE WHEN @empid IS NOT NULL THEN
209         N' AND empid = @eid' ELSE N'' END
210     + CASE WHEN @orderdate IS NOT NULL THEN
211         N' AND orderdate = @dt' ELSE N'' END;
212 EXEC sp_executesql
213     @stmt = @sql,
214     @params = N'@oid AS INT, @cid AS INT, @eid AS INT, @dt AS DATE',
215     @oid = @orderid,
216     @cid = @custid,
217     @eid = @empid,
218     @dt = @orderdate;
219 --end snippet dynamic_ds_02
220 GO
221
222 -- Test procedure
223 USE tempdb;
224 --start snippet dynamic_ds_test
225 EXEC dbo.GetOrders @orderdate = '20140101';
226 EXEC dbo.GetOrders @orderdate = '20140102';
227 EXEC dbo.GetOrders @orderid = 10248;
228 --end snippet dynamic_ds_test
229 GO
230
231
232 -- Find cached plans and their reuse
233 USE tempdb;

```

```

234  --start snippet cached_plans
235  SELECT usecounts, text
236  FROM sys.dm_exec_cached_plans AS CP
237       CROSS APPLY sys.dm_exec_sql_text(cp.plan_handle) AS ST
238  WHERE ST.text LIKE '%27702431-107C-478C-8157-6DFCECC148DD%'
239       AND ST.text NOT LIKE '%sys.dm_exec_cached_plans%'
240       AND CP.objtype = 'Prepared';
241  --end snippet cached_plans
242  GO

```

## Dynamic Sorting

```

244  -----
245  --Dynamic Sorting
246  -----
247
248  --Use static SQL with recompile
249  USE SP;
250  IF OBJECT_ID(N'dbo.GetSortedShippers', N'P') IS NOT NULL DROP PROC dbo.GetSortedShippers;
251  GO
252  --start snippet static_dsrt_rc
253  CREATE PROC dbo.GetSortedShippers
254      @colname AS sysname, @sortdir AS CHAR(1) = 'A'
255  AS
256
257  SELECT shipperid, companyname, phone
258  FROM Sales.Shippers
259  --end snippet static_dsrt_rc
260  --start snippet static_dsrt_rc_02
261  ORDER BY
262      CASE WHEN @colname = N'shipperid'   AND @sortdir = 'A'
263           THEN shipperid   END,
264      CASE WHEN @colname = N'companyname' AND @sortdir = 'A'
265           THEN companyname END,
266      CASE WHEN @colname = N'phone'       AND @sortdir = 'A'
267           THEN phone       END,
268      CASE WHEN @colname = N'shipperid'   AND @sortdir = 'D'
269           THEN shipperid   END DESC,
270      CASE WHEN @colname = N'companyname' AND @sortdir = 'D'
271           THEN companyname END DESC,
272      CASE WHEN @colname = N'phone'       AND @sortdir = 'D'
273           THEN phone       END DESC
274  OPTION (RECOMPILE);
275  --end snippet static_dsrt_rc_02
276  GO

```

```

277
278 -- Test procedure
279
280 --start snippet static_dsrt_rc_test
281 EXEC dbo.GetSortedShippers N'shipperid', N'D';
282 --end snippet static_dsrt_rc_test
283 GO
284
285 -- Use dynamic SQL
286 USE SP;
287 IF OBJECT_ID(N'dbo.GetSortedShippers', N'P') IS NOT NULL DROP PROC dbo.GetSortedShippers;
288 GO
289
290 --start snippet dynamic_dsrt
291 CREATE PROC dbo.GetSortedShippers
292     @colname AS sysname, @sortdir AS CHAR(1) = 'A'
293 AS
294 IF @colname NOT IN(N'shipperid', N'companyname', N'phone')
295     THROW 50001,
296         'Column name not supported. Possibly a SQL injection attempt.', 1;
297
298 DECLARE @sql AS NVARCHAR(1000);
299 SET @sql = N'SELECT shipperid, companyname, phone
300 FROM Sales.Shippers
301 ORDER BY '
302     + QUOTENAME(@colname) +
303     CASE @sortdir WHEN 'D' THEN N' DESC' ELSE '' END + ';';
304
305 EXEC sys.sp_executesql @stmt = @sql;
306 --end snippet dynamic_dsrt
307 GO
308
309 -- Test procedure
310 --start snippet dynamic_dsrt_test
311 EXEC dbo.GetSortedShippers N'shipperid', N'D';
312 --end snippet dynamic_dsrt_test
313 GO
314
315 -- Sort by multiple columns
316 USE SP;
317 IF OBJECT_ID(N'dbo.GetSortedShippers', N'P') IS NOT NULL DROP PROC dbo.GetSortedShippers;
318 GO
319
320 --start snippet dynamic_dsrt_mc_01
321 CREATE PROC dbo.GetSortedShippers
322     @colname1 AS sysname, @sortdir1 AS CHAR(1) = 'A',

```



```

323     @colname2 AS sysname = NULL, @sortdir2 AS CHAR(1) = 'A',
324     @colname3 AS sysname = NULL, @sortdir3 AS CHAR(1) = 'A'
325 AS
326 IF @colname1 NOT IN(N'shipperid', N'companyname', N'phone')
327     OR @colname2 IS NOT NULL
328     AND @colname2 NOT IN(N'shipperid', N'companyname', N'phone')
329     OR @colname3 IS NOT NULL
330     AND @colname3 NOT IN(N'shipperid', N'companyname', N'phone')
331     THROW 50001,
332     'Column name not supported. Possibly a SQL injection attempt.', 1;
333 --end snippet dynamic_dsrt_mc_01
334
335 --start snippet dynamic_dsrt_mc_02
336 DECLARE @sql AS NVARCHAR(1000);
337
338 SET @sql = N'SELECT shipperid, companyname, phone
339 FROM Sales.Shippers
340 ORDER BY '
341     + QUOTENAME(@colname1) +
342     CASE @sortdir1 WHEN 'D' THEN N' DESC' ELSE '' END
343     + ISNULL(N',' + QUOTENAME(@colname2) +
344     CASE @sortdir2 WHEN 'D' THEN N' DESC' ELSE '' END, N'')
345     + ISNULL(N',' + QUOTENAME(@colname3) +
346     CASE @sortdir3 WHEN 'D' THEN N' DESC' ELSE '' END, N'')
347     + ',';
348
349 EXEC sys.sp_executesql @stmt = @sql;
350 --end snippet dynamic_dsrt_mc_02
351 GO

```