

Intro to Database

Danny Tan

12/5/2019

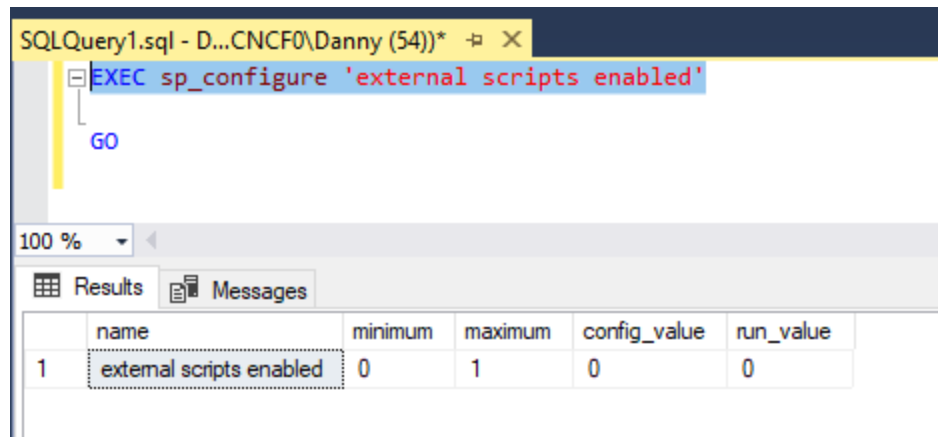
Python in-database Analytics

- Reduce data movement
- Easy Deployment
- Enterprise-grade equipment

Setup/Configuration

EXEC sp_configure 'external scripts enabled'

GO

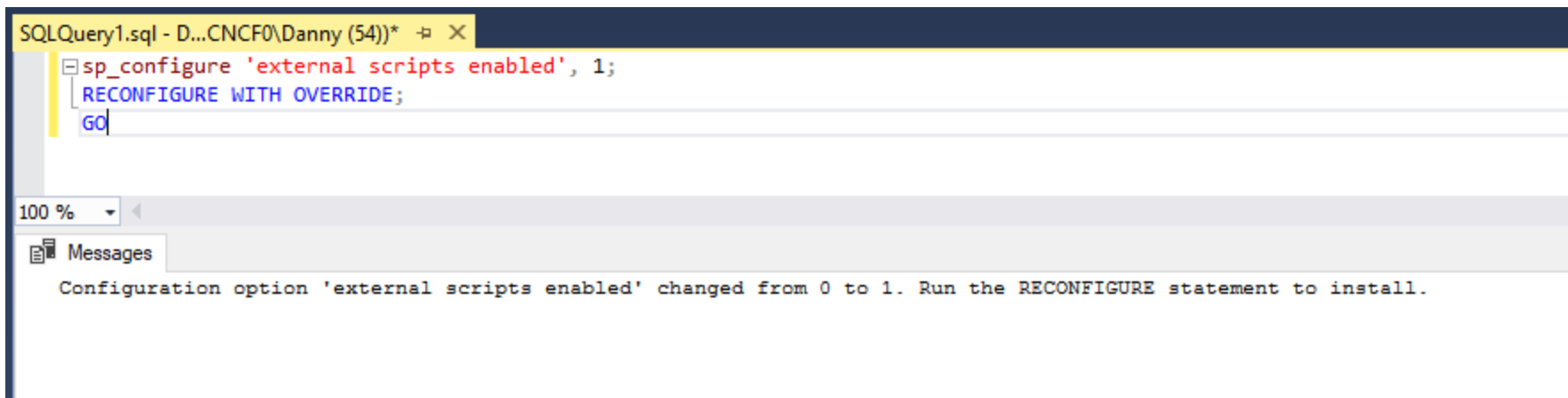


The screenshot shows a SQL Server Enterprise Manager window with a query editor at the top containing the command `EXEC sp_configure 'external scripts enabled'` followed by `GO`. Below the editor, the 'Results' tab is active, displaying a table with the configuration state of 'external scripts enabled'.

	name	minimum	maximum	config_value	run_value
1	external scripts enabled	0	1	0	0

Setup/Configuration

```
sp_configure 'external scripts enabled', 1;  
RECONFIGURE WITH OVERRIDE;  
GO
```



The screenshot shows a SQL Server Enterprise Manager window titled "SQLQuery1.sql - D...CNCFO\Danny (54))*". The query editor contains the following T-SQL code:

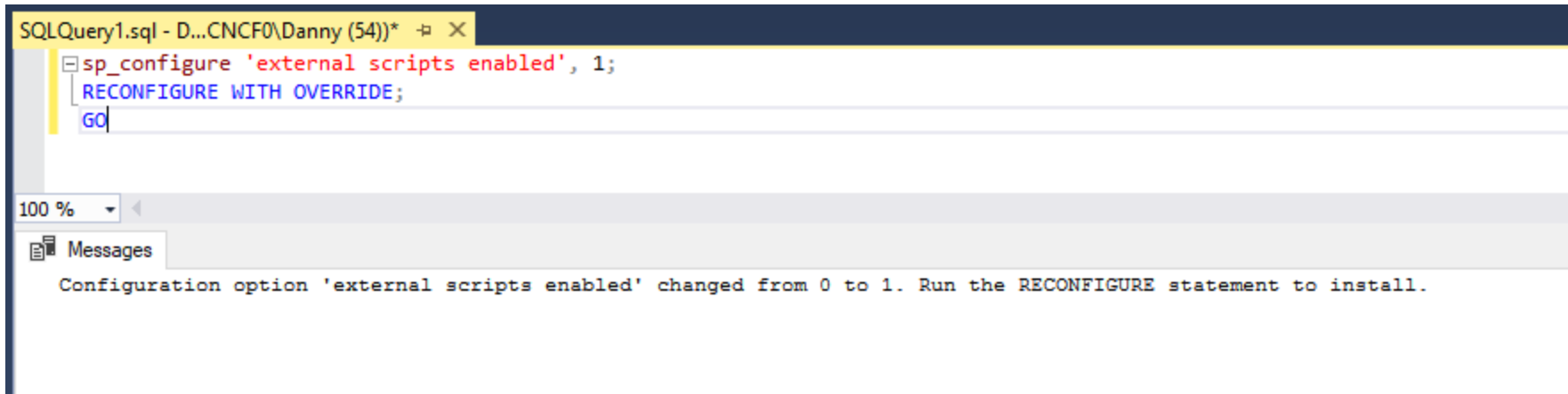
```
sp_configure 'external scripts enabled', 1;  
RECONFIGURE WITH OVERRIDE;  
GO
```

Below the query editor, the "Messages" pane shows the following output:

```
Configuration option 'external scripts enabled' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

Setup/Configuration

```
sp_configure 'external scripts enabled', 1;  
RECONFIGURE WITH OVERRIDE;  
GO
```



The screenshot displays the SQL Server Enterprise Manager interface. At the top, a query window titled 'SQLQuery1.sql - D...CNCFO\Danny (54))' contains the following T-SQL code:

```
sp_configure 'external scripts enabled', 1;  
RECONFIGURE WITH OVERRIDE;  
GO
```

Below the query window, a 'Messages' window shows the execution result:

```
Configuration option 'external scripts enabled' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

Setup/Configuration

After executing this command, restart the SQL Server service in order to use this feature. Once this feature gets enabled, we can use this procedure to execute external scripts on the server.

Python In-Database Format

```
sp_execute_external_script
```

```
    @language = N'language',
```

```
    @script = N'script'
```

```
    [ , @input_data_1 = N'input_data_1' ]
```

```
    [ , @input_data_1_name = N'input_data_1_name' ]
```

```
    [ , @output_data_1_name = N'output_data_1_name' ]
```

```
    [ , @parallel = 0 | 1 ]
```

```
    [ , @params = N'@parameter_name data_type [ OUT |  
OUTPUT ] [ ,...n ]' ]
```

```
    [ , @parameter1 = 'value1' [ OUT | OUTPUT ] [ ,...n ] ]
```

Examples

```
execute sp_execute_external_script
@language = N'Python',
@script = N'
l = [15, 10, 22, 36, 5, 4, 3, 2, 1]
print(sum(l) / float(len(l)))
'
```

```
execute sp_execute_external_script
@language = N'Python',
@script = N'
for i in range(5):
    if i<3 :
        print("i is now:", i*2)'
```


Sample Table

```
select * from sample_stock
```

100 %



Results



Messages

	date1	cusip	open_close	adj_close
1	2011-09-01 00:00:00.000	abc	100	102
2	2011-09-02 00:00:00.000	abc	100	105
3	2011-09-03 00:00:00.000	abc	100	107
4	2011-09-04 00:00:00.000	abc	100	108
5	2011-09-01 00:00:00.000	f	100	80
6	2011-09-02 00:00:00.000	f	100	76
7	2011-09-03 00:00:00.000	f	100	80

Sample Python Code

```
EXEC sp_execute_external_script
```

```
@language = N'Python',
```

```
@script = N'
```

```
OutputDataSet = InputDataSet
```

```
OutputDataSet["Double"] = pandas.Series([i*2 for i in InputDataSet["adj_close"]], index = InputDataSet.index,  
dtype = "float")
```

```
'
```

```
@input_data_1 = N'SELECT adj_close FROM sample_stock'
```

```
WITH RESULT SETS (("Input" nvarchar(10) null,"Double" nvarchar(10) null))
```

Use Python File

```
EXEC sp_execute_external_script
```

```
@language = N'Python',
```

```
@script = N'
```

```
from SQLPython import price_double
```

```
OutputDataSet = price_double(InputDataSet)
```

```
;',
```

```
@input_data_1 = N'SELECT adj_close FROM sample_stock'
```

```
WITH RESULT SETS (("Input" nvarchar(10) null,"Double" nvarchar(10)  
null))
```

SQLPython.py x

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 13 20:30:56 2018

@author: Danny
"""

def price_double(InputDataSet):
    import pandas
    OutputDataSet = InputDataSet

    OutputDataSet["Double"] = pandas.Series([i*2 for i in InputDataSet["adj_close"]], index = InputDataSet.index, dtype = "float")

    return OutputDataSet
```



init.py x

```
__author__ = "Danny Tan"
__version__ = "1.0.0"

from .SQLPython import price_double
```

w

Program Files > Microsoft SQL Server > MSSQL14.SQL_EXPRESS2017 > PYTHON_SERVICES > SQLPython

	Name	Date modified	Type	Size
	__init__.py	10/13/2018 9:13 PM	PY File	1 KB
	SQLPython.py	10/13/2018 9:13 PM	PY File	1 KB

SQLQuery3.sql - D...CNCFO\Danny (52))* X SQLQuery1.sql - D...CNCFO\Danny (51))* SQLQuery4.sql - D...CNCFO\Danny (59))* SQLQuery2.sql -

```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
from SQLPython import price_double
OutputDataSet = price_double(InputDataSet)
',
@input_data_1 = N'SELECT adj_close FROM sample_stock'
WITH RESULT SETS (('Input' nvarchar(10) null,"Double" nvarchar(10) null))
```

100 %

Results Messages

	Input	Double
1	102	204
2	105	210
3	107	214
4	108	216
5	80	160
6	76	152
7	80	160