

Nada to Speaker: Demystifying and Diminishing Dynamic SQL Difficulties

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Last Thursday I spoke at NTSSUG (North Texas SQL Server User Group). This is my local group (only 1.5 hours away from me) and my second time speaking for them. I don't speak frequently so I'm always more than a bit nervous, but the crowd was great, laughed in the right spots, asked some really good questions and all around had some great energy. This post is primarily to share the Powerpoint and scripts I used during my demo but since I don't like doing a post that is just links to some files I'm adding some of the content below as well.

One of the ideas that I mentioned is that there are two types of Dynamic SQL as I see it. The processes that I use to create each are pretty similar but of course not quite the same.

One to Many

Using a single piece of code (typically a query) to generate multiple lines of SQL.

Example

```
SELECT 'ALTER DATABASE ' + QUOTENAME(name) +
    'SET COMPATIBILITY_LEVEL = 130'
FROM sys.databases
WHERE compatibility_level < 130
```

Process

1. Write the SQL statement first.
2. Write the query that pulls the result set you want.

Nada to Speaker: Demystifying and Diminishing Dynamic SQL Difficulties

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Last Thursday I spoke at NTSSUG (North Texas SQL Server User Group). This is my local group (only 1.5 hours away from me) and my second time speaking for them. I don't speak frequently so I'm always more than a bit nervous, but the crowd was great, laughed in the right spots, asked some really good questions and all around had some great energy. This post is primarily to share the Powerpoint and scripts I used during my demo but since I don't like doing a post that is just links to some files I'm adding some of the content below as well.

One of the ideas that I mentioned is that there are two types of Dynamic SQL as I see it. The processes that I use to create each are pretty similar but of course not quite the same.

One to Many

Using a single piece of code (typically a query) to generate multiple lines of SQL.

Example

```
SELECT 'ALTER DATABASE ' + QUOTENAME(name) +
    'SET COMPATIBILITY_LEVEL = 130'
FROM sys.databases
WHERE compatibility_level < 130
```

Process

1. Write the SQL statement first.
2. Write the query that pulls the result set you want.

Nada to Speaker: Demystifying and Diminishing Dynamic SQL Difficulties

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Last Thursday I spoke at NTSSUG (North Texas SQL Server User Group). This is my local group (only 1.5 hours away from me) and my second time speaking for them. I don't speak frequently so I'm always more than a bit nervous, but the crowd was great, laughed in the right spots, asked some really good questions and all around had some great energy. This post is primarily to share the Powerpoint and scripts I used during my demo but since I don't like doing a post that is just links to some files I'm adding some of the content below as well.

One of the ideas that I mentioned is that there are two types of Dynamic SQL as I see it. The processes that I use to create each are pretty similar but of course not quite the same.

One to Many

Using a single piece of code (typically a query) to generate multiple lines of SQL.

Example

```
SELECT 'ALTER DATABASE ' + QUOTENAME(name) +
    'SET COMPATIBILITY_LEVEL = 130'
FROM sys.databases
WHERE compatibility_level < 130
```

Process

1. Write the SQL statement first.
2. Write the query that pulls the result set you want.

Nada to Speaker: Demystifying and Diminishing Dynamic SQL Difficulties

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Last Thursday I spoke at NTSSUG (North Texas SQL Server User Group). This is my local group (only 1.5 hours away from me) and my second time speaking for them. I don't speak frequently so I'm always more than a bit nervous, but the crowd was great, laughed in the right spots, asked some really good questions and all around had some great energy. This post is primarily to share the Powerpoint and scripts I used during my demo but since I don't like doing a post that is just links to some files I'm adding some of the content below as well.

One of the ideas that I mentioned is that there are two types of Dynamic SQL as I see it. The processes that I use to create each are pretty similar but of course not quite the same.

One to Many

Using a single piece of code (typically a query) to generate multiple lines of SQL.

Example

```
SELECT 'ALTER DATABASE ' + QUOTENAME(name) +
    'SET COMPATIBILITY_LEVEL = 130'
FROM sys.databases
WHERE compatibility_level < 130
```

Process

1. Write the SQL statement first.
2. Write the query that pulls the result set you want.

Nada to Speaker: Demystifying and Diminishing Dynamic SQL Difficulties

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Last Thursday I spoke at NTSSUG (North Texas SQL Server User Group). This is my local group (only 1.5 hours away from me) and my second time speaking for them. I don't speak frequently so I'm always more than a bit nervous, but the crowd was great, laughed in the right spots, asked some really good questions and all around had some great energy. This post is primarily to share the Powerpoint and scripts I used during my demo but since I don't like doing a post that is just links to some files I'm adding some of the content below as well.

One of the ideas that I mentioned is that there are two types of Dynamic SQL as I see it. The processes that I use to create each are pretty similar but of course not quite the same.

One to Many

Using a single piece of code (typically a query) to generate multiple lines of SQL.

Example

```
SELECT 'ALTER DATABASE ' + QUOTENAME(name) +
    'SET COMPATIBILITY_LEVEL = 130'
FROM sys.databases
WHERE compatibility_level < 130
```

Process

1. Write the SQL statement first.
2. Write the query that pulls the result set you want.

Nada to Speaker: Demystifying and Diminishing Dynamic SQL Difficulties

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Last Thursday I spoke at NTSSUG (North Texas SQL Server User Group). This is my local group (only 1.5 hours away from me) and my second time speaking for them. I don't speak frequently so I'm always more than a bit nervous, but the crowd was great, laughed in the right spots, asked some really good questions and all around had some great energy. This post is primarily to share the Powerpoint and scripts I used during my demo but since I don't like doing a post that is just links to some files I'm adding some of the content below as well.

One of the ideas that I mentioned is that there are two types of Dynamic SQL as I see it. The processes that I use to create each are pretty similar but of course not quite the same.

One to Many

Using a single piece of code (typically a query) to generate multiple lines of SQL.

Example

```
SELECT 'ALTER DATABASE ' + QUOTENAME(name) +
    'SET COMPATIBILITY_LEVEL = 130'
FROM sys.databases
WHERE compatibility_level < 130
```

Process

1. Write the SQL statement first.
2. Write the query that pulls the result set you want.

Nada to Speaker: Demystifying and Diminishing Dynamic SQL Difficulties

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Last Thursday I spoke at NTSSUG (North Texas SQL Server User Group). This is my local group (only 1.5 hours away from me) and my second time speaking for them. I don't speak frequently so I'm always more than a bit nervous, but the crowd was great, laughed in the right spots, asked some really good questions and all around had some great energy. This post is primarily to share the Powerpoint and scripts I used during my demo but since I don't like doing a post that is just links to some files I'm adding some of the content below as well.

One of the ideas that I mentioned is that there are two types of Dynamic SQL as I see it. The processes that I use to create each are pretty similar but of course not quite the same.

One to Many

Using a single piece of code (typically a query) to generate multiple lines of SQL.

Example

```
SELECT 'ALTER DATABASE ' + QUOTENAME(name) +
    'SET COMPATIBILITY_LEVEL = 130'
FROM sys.databases
WHERE compatibility_level < 130
```

Process

1. Write the SQL statement first.
2. Write the query that pulls the result set you want.