

## Search Results for: all your database

### All Your Database Are Belong to Us

July 13, 2014 By [Allen Underwood](#) – 8 Comments



Podcast: [Play in new window](#) | [Download](#)

Subscribe: [Apple Podcasts](#) | [Android](#) | [Google Podcasts](#) | [Stitcher](#) | [Tuneln](#) | [Spotify](#) | [RSS](#)

Part one of our two part database podcast starts with choosing the RDBMS (Relational Database Management System) and what to do when you run into deficiencies in that particular database system. First and foremost, what's with the title?! Are these guys grammatically challenged? If that was your first thought, then you should check out this link: [http://en.wikipedia.org/wiki/All\\_your\\_base\\_are\\_belong\\_to\\_us](http://en.wikipedia.org/wiki/All_your_base_are_belong_to_us) Others who have been around a little while (longer than us three 21 year olds), we hope you got a kick out of the title. [\[Read more\]](#) 1

#### Subscribe to Our Newsletter

Name

Email

☐

I'm not a robot



#### Infotainment at it's finest

Like what you see? Check out the [Coding Blocks Podcast](#) on your next commute!



#### JetBrains



[Allen Underwood](#)

<https://www.codingblocks.net/>

Twitter: [@theallenu](#)

<https://www.linkedin.com/in/allenunderwood/>

<https://github.com/codingblocks/>

# Boost Your MSSQL Productivity

---



# Docker – If You Don't Have It

---

Go download it!!!

<https://www.docker.com/products/docker-desktop>

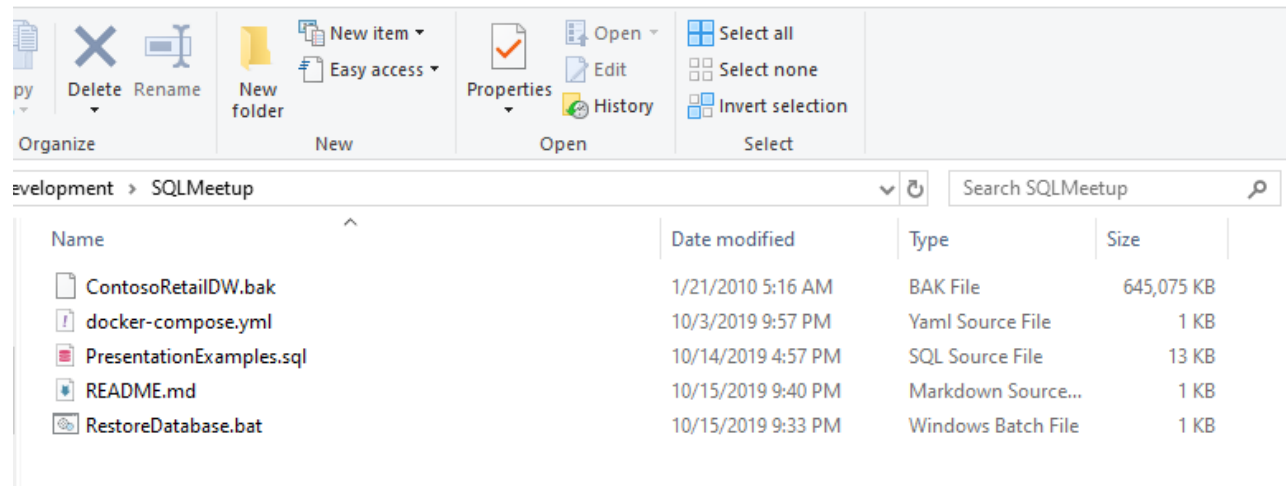
For the best experience to play along with the examples in this slide-deck, you'll want to be on Windows 10 Professional

# Docker – BEFORE You Get Started

If you want to play around in the SQL file that's included in this repository, you'll need to download the ContosoRetailDW file into the SAME directory as the docker-compose.yml (included in the root folder of this repository)

<https://www.microsoft.com/en-us/download/details.aspx?id=18279>

If you cloned the repo, that folder should look like this (after you download the file above)



# Docker – Getting Started

---

Take a look at the docker-compose.yml file in this repository

- You'll see three services in there – all three SQL Server instances

From the command line, in the same directory where the docker-compose.yml file exists, run the following (make sure you're using in Linux container mode for Docker)

```
docker-compose up -d
```

You should see something similar to this...

```
D:\Development\SQLMeetup
λ docker-compose up -d
Creating network "sqlmeetup_default" with the default driver
Creating sql-server ... done
Creating sql-stage  ... done
Creating sql-prod   ... done
D:\Development\SQLMeetup
λ
```

# Restoring the ContosoRetailDW

You will need to run the RestoreDatabase.bat file – it will restore the ContosoRetailDW into the “dev” environment → Port 1500 on the next slide

- This could take a few minutes – be patient – it’s restoring a 1.3GB database 😊

**RestoreDatabase.bat**

```
D:\Development\SQLMeetup
λ RestoreDatabase.bat

docker-compose exec sql-dev /opt/mssql-tools/bin/sqlcmd -U sa -P c0dingbl0cks! -H localhost -q "RESTORE DATABASE [ContosoRetailDW] FROM DISK = N'/var/opt/mssql/data/ContosoRetailDW.bak' WITH FILE = 1, MOVE N'ContosoRetailDW2.0' TO N'/var/opt/mssql/data/ContosoRetailDW.mdf', MOVE N'ContosoRetailDW2.0_log' TO N'/var/opt/mssql/data/ContosoRetailDW.ldf', NOUNLOAD, STATS = 5"

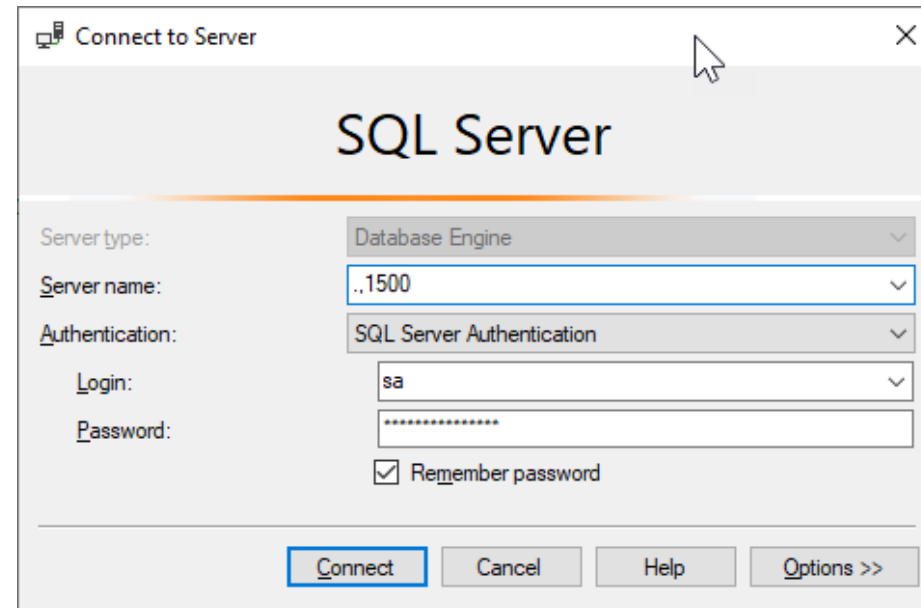
5 percent processed.
10 percent processed.
15 percent processed.
20 percent processed.

Database 'ContosoRetailDW' running the upgrade step from version 863 to version 864.
Database 'ContosoRetailDW' running the upgrade step from version 864 to version 865.
Database 'ContosoRetailDW' running the upgrade step from version 865 to version 866.
Database 'ContosoRetailDW' running the upgrade step from version 866 to version 867.
Database 'ContosoRetailDW' running the upgrade step from version 867 to version 868.
Database 'ContosoRetailDW' running the upgrade step from version 868 to version 869.
RESTORE DATABASE successfully processed 157986 pages in 32.330 seconds (38.176 MB/sec).
```

# Now that SQL Server is Started with Data

Open up SSMS and try to connect

- If you didn't change anything in the docker-compose.yml file, enter the following – notice that server name is period comma (period is an alias for localhost) – note that changing the port in the Server Name field represents one of the three services from the docker-compose.yml file
- 1500 → Development (ContosoRetailDW)
- 1501 → Staging (Empty)
- 1502 → Production (Empty)
  - Server Name: .,1500
  - Authentication: Sql Server
  - Login: sa
  - Password: c0dingbl0cks!



Connect to Server

SQL Server

Server type: Database Engine

Server name: .,1500

Authentication: SQL Server Authentication

Login: sa

Password: \*\*\*\*\*

☒ Remember password

Connect Cancel Help Options >>



# Open Up PresentationExamples.sql

---

Now it's time to follow along – PresentationExamples.sql is in the root folder of this project

Most of the slides in this presentation have an accompanying query or set of queries in the sql file. Open that thing up and play along.



# Setting the Bar...

---

INNER JOIN → JOIN

LEFT OUTER JOIN → LEFT JOIN

~~RIGHT OUTER JOIN → RIGHT JOIN~~

FULL OUTER JOIN → OUTER JOIN

CROSS JOIN

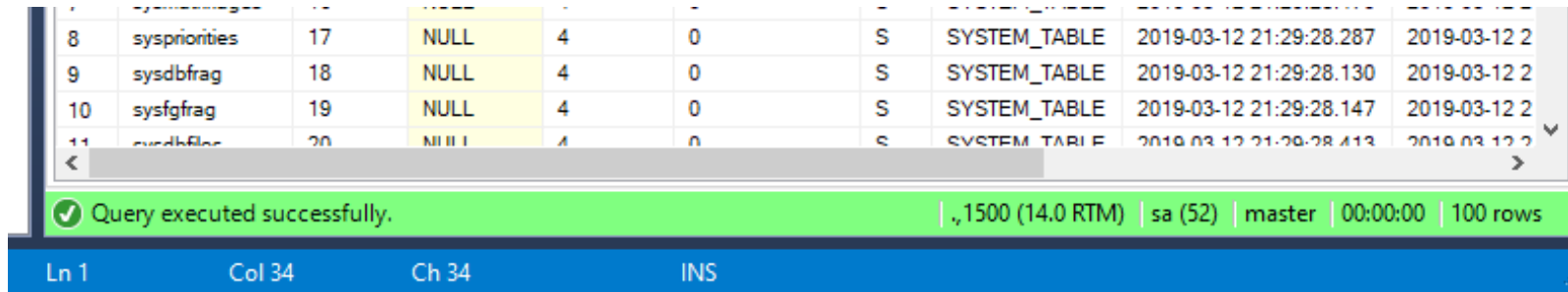
CROSS APPLY

OUTER APPLY

# SSMS Tips & Tricks

---

Color code your query windows by connection



8	syspriorities	17	NULL	4	0	S	SYSTEM_TABLE	2019-03-12 21:29:28.287	2019-03-12 2
9	sysdbfrag	18	NULL	4	0	S	SYSTEM_TABLE	2019-03-12 21:29:28.130	2019-03-12 2
10	sysfgfrag	19	NULL	4	0	S	SYSTEM_TABLE	2019-03-12 21:29:28.147	2019-03-12 2
11	sysdbfrag	20	NULL	4	0	S	SYSTEM_TABLE	2019-03-12 21:29:28.113	2019-03-12 2

✓ Query executed successfully. | ~,1500 (14.0 RTM) | sa (52) | master | 00:00:00 | 100 rows

Ln 1 Col 34 Ch 34 INS

# SSMS Tips & Tricks

Color code your query windows by connection

The screenshot displays two overlapping query windows in SQL Server Enterprise Manager. The top window has a green header bar and a status bar indicating it is connected to 'sa (52)' on the 'master' server. The bottom window has a yellow header bar and a status bar indicating it is connected to 'sa (55)' on the 'master' server. Both windows show a table of system information with columns for object name, type, and other details. The status bars also show the number of rows returned (100 rows) and the execution time (00:00:00).

Object Name	Type	Other Columns
syspriorities	17	NULL
sysdbfrag	18	NULL
sysfgfrag	19	NULL
sysdbfrag	20	NULL

# SSMS Tips & Tricks

Color code your query windows by connection

The image displays three overlapping query windows from SQL Server Enterprise Manager, each showing a different connection. The windows are color-coded: green, yellow, and red. Each window displays a table of system information and a status bar at the bottom.

**Green Window (Connection: sa (52))**

Ln	Col	Ch
8	syspriorities	17
9	sysdbfrag	18
10	sysfgfrag	19
11	sysdbfrag	20

Query executed successfully. | ,1500 (14.0 RTM) | sa (52) | master | 00:00:00 | 100 rows

Ln 1 Col 34 Ch 34

**Yellow Window (Connection: sa (55))**

Ln	Col	Ch
10	sysfgfrag	19
11	sysdbfrag	20

Query executed successfully. | ,1501 (14.0 RTM) | sa (55) | master | 00:00:00 | 100 rows

Ln 1 Col 34

**Red Window (Connection: sa (51))**

Ln	Col	Ch
7	sysmatrixages	16
8	syspriorities	17
9	sysdbfrag	18
10	sysfgfrag	19
11	sysdbfrag	20

Query executed successfully. | ,1502 (14.0 RTM) | sa (51) | master | 00:00:00 | 108 rows

Ln 1 Col 26 Ch 26 INS

# Highlight Object Name, then Alt + F1

---

Find out about database objects with this quick shortcut

- Tables
- Stored Procedures
- Functions

# Dry Run Your Queries

---

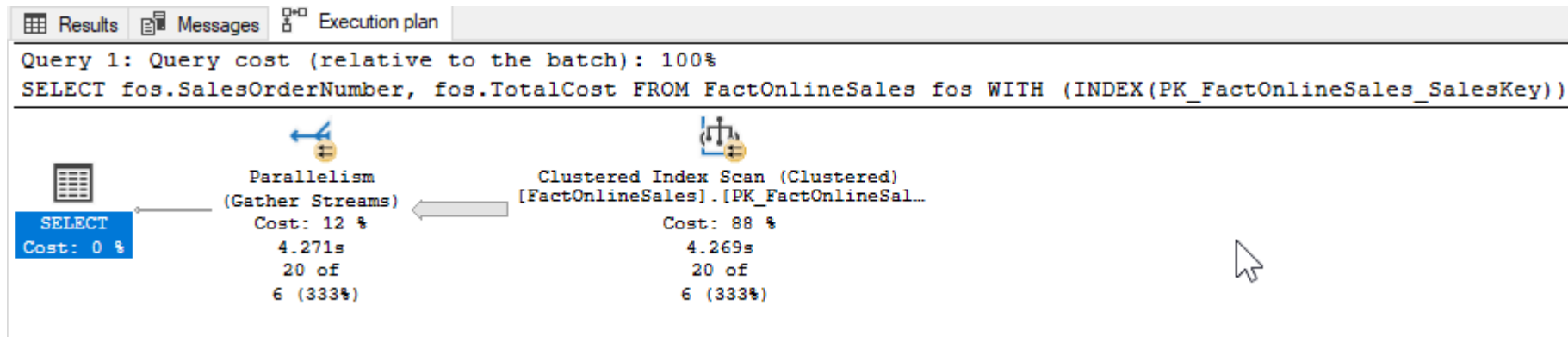
```
UPDATE c SET  
    FirstName = 'Elisabeth'  
FROM DimCustomer c
```

```
-- UPDATE c SET  
SELECT c.*,  
    FirstName = 'Elisabeth'  
FROM DimCustomer c
```

# Get Familiar with Execution Plans

```
SELECT
    fos.SalesOrderNumber,
    fos.TotalCost
FROM FactOnlineSales fos WITH (INDEX(PK_FactOnlineSales_SalesKey))
WHERE SalesOrderNumber IN ('20070101311332', '20070101711340')
```

**Clustered Index Scan → NOT what you want to see most of the time**





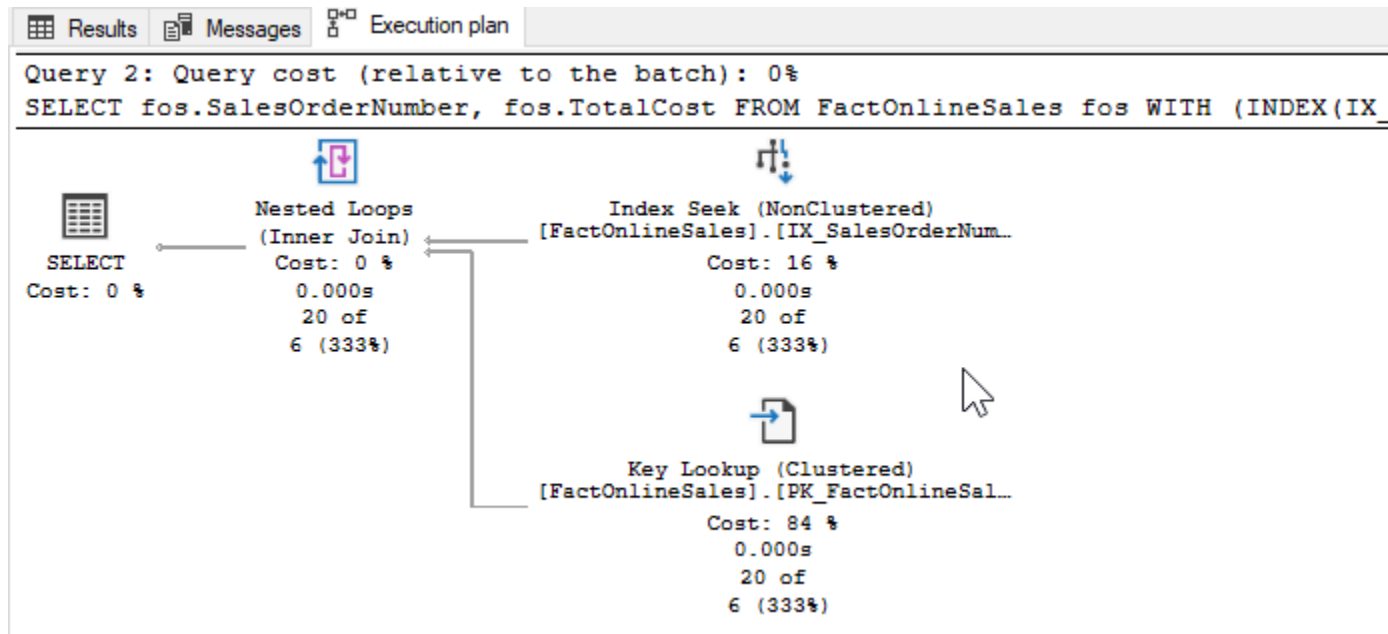
# Key Lookup – What's That?

SELECT

fos.SalesOrderNumber,  
fos.TotalCost

FROM FactOnlineSales fos WITH (INDEX (IX\_SalesOrderNumber))

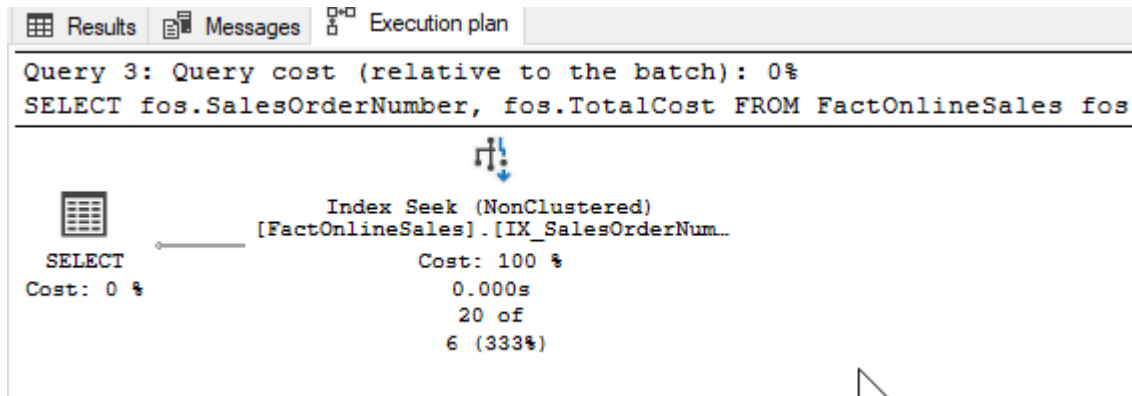
WHERE SalesOrderNumber IN ('20070101311332', '20070101711340')



CREATE INDEX IX\_SalesOrderNumber ON  
dbo.FactOnlineSales(  
SalesOrderNumber  
)  
GO

# Index Seek → Pay Dirt!

```
SELECT
    fos.SalesOrderNumber,
    fos.TotalCost
FROM FactOnlineSales fos WITH (INDEX (IX_SalesOrderNumberWithCost))
WHERE SalesOrderNumber IN ('20070101311332', '20070101711340')
```



```
CREATE INDEX IX_SalesOrderNumberWithCost ON
dbo.FactOnlineSales(
    SalesOrderNumber
)
INCLUDE (
    TotalCost
)
GO
```

# sp\_who2 Hidden Proc

EXEC sp\_who2 'active'

SQLQuery4.sql - ,1500.master (sa (52))\* X PresentationExample...(51) Executing...\*

SQLQuery2.sql - not connected

EXEC sp\_who2 'active'

100 %

Results Messages

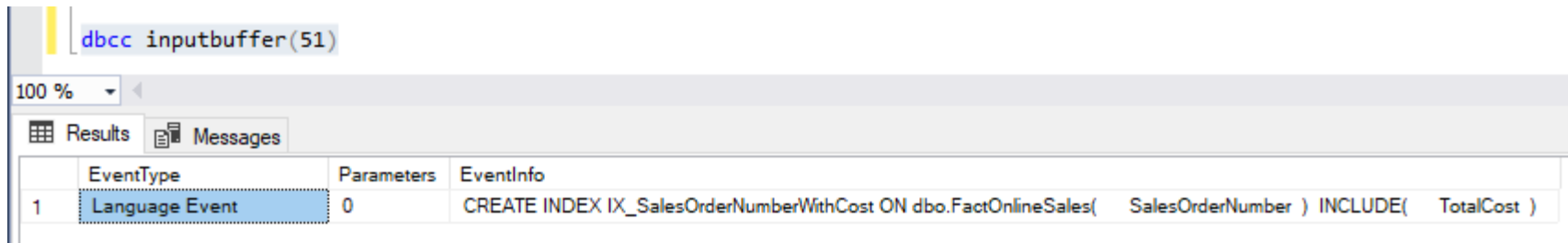
SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramName	SPID	REQUESTID	
4	4	BACKGROU...	sa	.	.	NULL	LAZY WRITER	878000	0	10/13 14:55:...	4	0	
5	5	BACKGROU...	sa	.	.	NULL	LOCK MONITOR	171000	0	10/13 14:55:...	5	0	
6	6	BACKGROU...	sa	.	.	master	SIGNAL HANDLER	0	0	10/13 14:55:...	6	0	
7	7	BACKGROU...	sa	.	.	NULL	XIO_LEASE_RENEWA	395000	0	10/13 14:55:...	7	0	
8	8	BACKGROU...	sa	.	.	master	BRKR TASK	0	0	10/13 14:55:...	8	0	
9	9	BACKGROU...	sa	.	.	NULL	XIO_RETRY_WORKER	425000	0	10/13 14:55:...	9	0	
10	10	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 18:04:...	10	0	
11	11	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 18:04:...	11	0	
12	12	BACKGROU...	sa	.	.	NULL	RESOURCE MONITOR	139960	0	10/13 14:55:...	12	0	
13	13	BACKGROU...	sa	.	.	NULL	XE TIMER	743997	0	10/13 14:55:...	13	0	
14	14	BACKGROU...	sa	.	.	NULL	XE DISPATCHER	16999	0	10/13 14:55:...	14	0	
15	15	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 12:44:...	15	0	
16	16	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/13 14:55:...	16	0	
17	17	BACKGROU...	sa	.	.	master	TRACE QUEUE TASK	48000	0	10/13 14:55:...	17	0	
18	18	BACKGROU...	sa	.	.	NULL	SYSTEM_HEALTH_MO	14999	0	10/13 14:55:...	18	0	
19	19	BACKGROU...	sa	.	.	NULL	RECEIVE	30303	0	10/13 14:55:...	19	0	
20	20	BACKGROU...	sa	.	.	master	BRKR TASK	0	0	10/13 14:55:...	20	0	
21	21	BACKGROU...	sa	.	.	master	TASK MANAGER	0	0	10/13 14:55:...	21	0	
22	22	BACKGROU...	sa	.	.	master	CHECKPOINT	17000	96	10/13 14:55:...	22	0	
23	23	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 12:44:...	23	0	
24	24	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 12:29:...	24	0	
25	25	BACKGROU...	sa	.	.	NULL	HADR_AR_MGR_NOTI	0	0	10/13 14:55:...	25	0	
26	26	BACKGROU...	sa	.	.	master	BRKR EVENT HNDLR	1000	103	10/13 14:55:...	26	0	
27	27	BACKGROU...	sa	.	.	master	BRKR TASK	264998	0	10/13 14:55:...	27	0	
28	28	BACKGROU...	sa	.	.	master	BRKR TASK	685000	0	10/13 14:55:...	28	0	
29	29	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 12:44:...	29	0	
30	30	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 18:04:...	30	0	
31	31	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 18:04:...	31	0	
32	32	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 12:44:...	32	0	
33	33	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 18:04:...	33	0	
34	34	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 18:04:...	34	0	
35	35	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 18:06:...	35	0	
36	36	sleeping	sa	.	.	master	TASK MANAGER	0	0	10/14 12:41:...	36	0	
37	51	SUSPENDED	sa	DESKT...	.	Conto...	CREATE INDEX	0	4	10/14 18:08:...	Microsoft S...	51	0
38	51	RUNNABLE		DESKT...	.	Conto...	CREATE INDEX	420000	0	10/14 18:08:...	Microsoft S...	51	0
39	51	RUNNABLE		DESKT...	.	Conto...	CREATE INDEX	443000	0	10/14 18:08:...	Microsoft S...	51	0
40	51	RUNNABLE		DESKT...	.	Conto...	CREATE INDEX	448000	0	10/14 18:08:...	Microsoft S...	51	0
41	51	RUNNABLE		DESKT...	.	Conto...	CREATE INDEX	422000	0	10/14 18:08:...	Microsoft S...	51	0
42	52	RUNNABLE	sa	DESKT...	.	master	SELECT INTO	4000	8	10/14 18:08:...	Microsoft S...	52	0

Query executed successfully.

# Now...DBCC INPUTBUFFER(spид)

Use the SPID from the previous slide, EXEC sp\_who2 results to find information about anything that looks like it needs to be investigated...

`dbcc inputbuffer(51)`



The screenshot shows the SQL Server Enterprise Manager interface. The command `dbcc inputbuffer(51)` is entered in the command window. Below the command window, the 'Results' tab is active, displaying a table with the following data:

	EventType	Parameters	EventInfo
1	Language Event	0	CREATE INDEX IX_SalesOrderNumberWithCost ON dbo.FactOnlineSales( SalesOrderNumber ) INCLUDE( TotalCost )

# Dark Theme Anyone?

---

## **SSMS 2016**

C:\Program Files (x86)\Microsoft SQL Server\130\Tools\Binn\ManagementStudio

## **SSMS 17**

C:\Program Files (x86)\Microsoft SQL Server\140\Tools\Binn\ManagementStudio

## **SSMS 18**

C:\Program Files (x86)\Microsoft SQL Server Management Studio 18\Common7\IDE

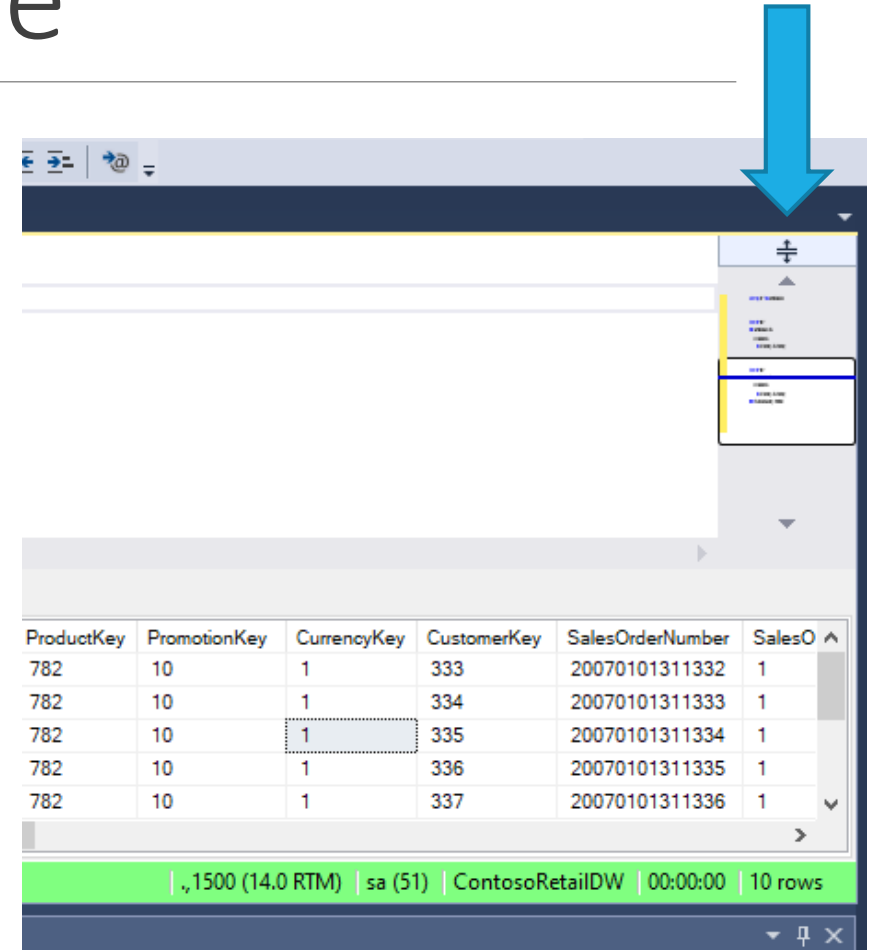
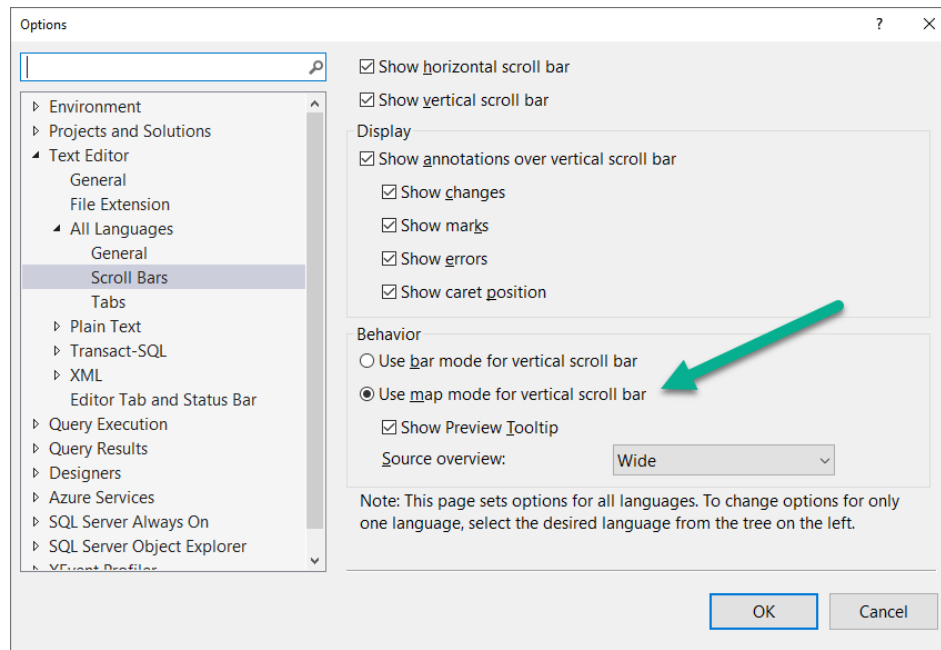
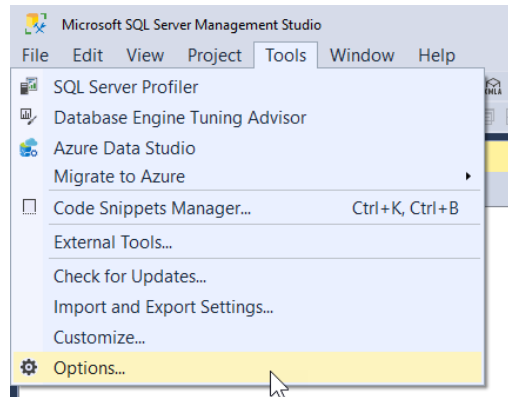
## **Comment out Dark Theme in ssms.pkgundef file**

// Remove Dark theme

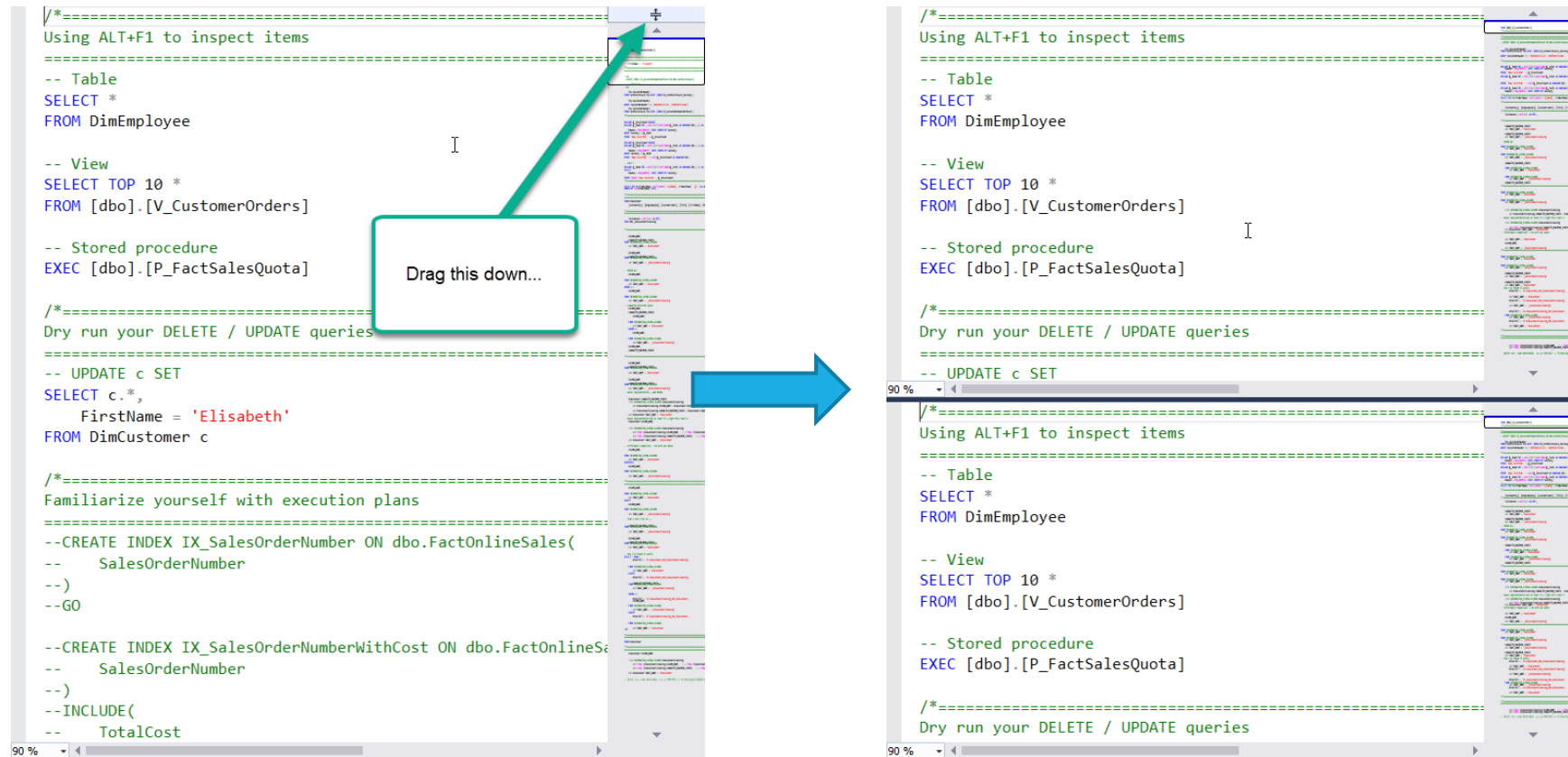
//[\$RootKey\$\Themes\{1ded0138-47ce-435e-84ef-9ec1f439b749}]



# Vertical Scrollbar Map Mode



# Quick Vertical Split





# Using CONCAT – Simplify Your Code

---

```
DECLARE @__ResultCount BIGINT
DECLARE @__Seed BIGINT = DATEDIFF(s, '1970-01-01 00:00:00', SYSDATETIME())
DECLARE @__Rand INT = CAST(RIGHT(CAST(RAND(@__Seed) AS VARCHAR(200)), 2) AS INT)

SELECT
    *,
    Rownum = ROW_NUMBER() OVER (ORDER BY SalesKey)
FROM FactSales
WHERE SalesKey <= @__Rand

SET @__ResultCount = @@ROWCOUNT

PRINT CONCAT('Rows Selected: ', @__ResultCount)
```

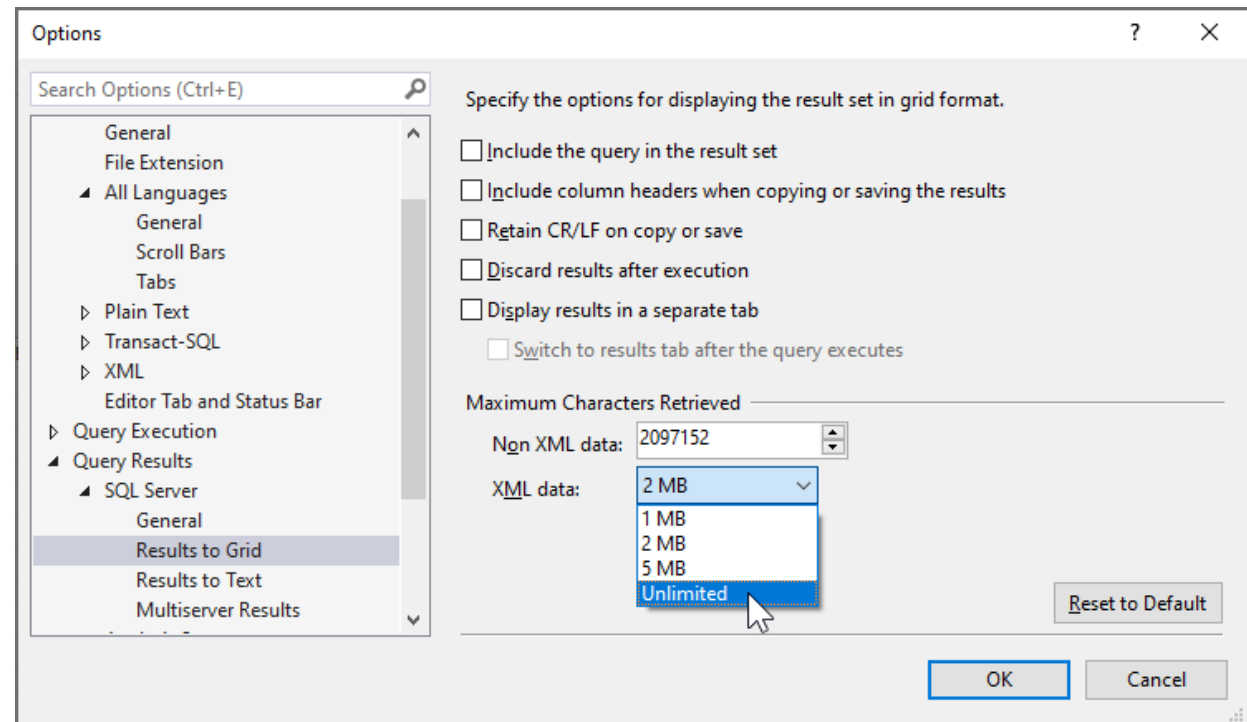
# Retrieving Large Text Blocks

```
SELECT TOP 10 ProductName,  
CAST(CONCAT('<![CDATA[' , ProductName, ']]>') AS  
XML) ProductNameAsXML  
FROM DimProduct  
ORDER BY LEN(ProductName) DESC
```

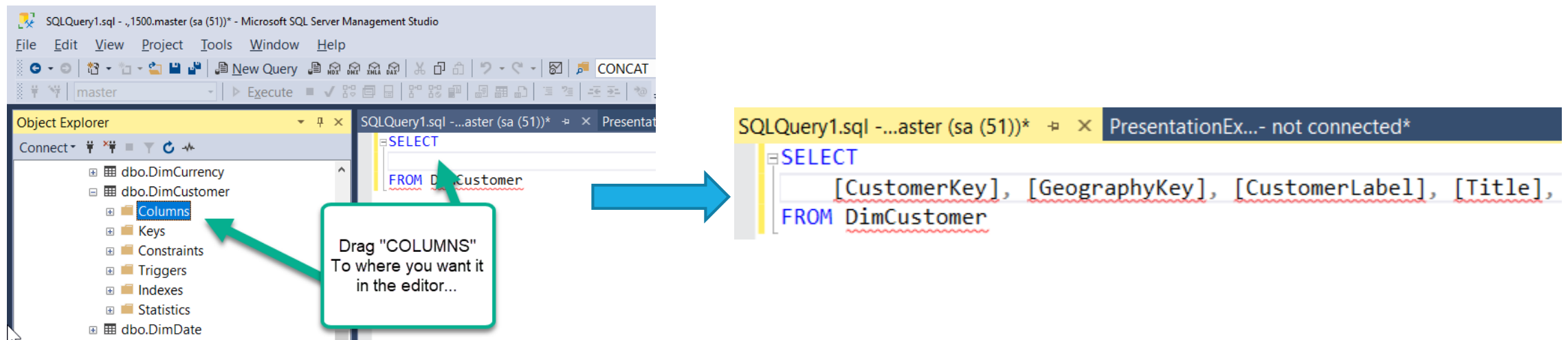
- The beauty is you can set XML output to no size limit
- You need the CDATA so that XML characters don't blow up the CAST

Maximum NON-XML data size

- 2,097,152 characters

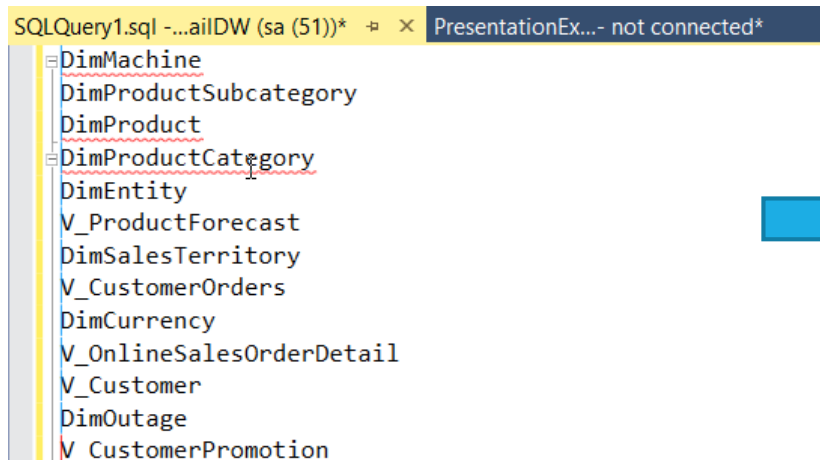


# Drag n' Drop Some Things

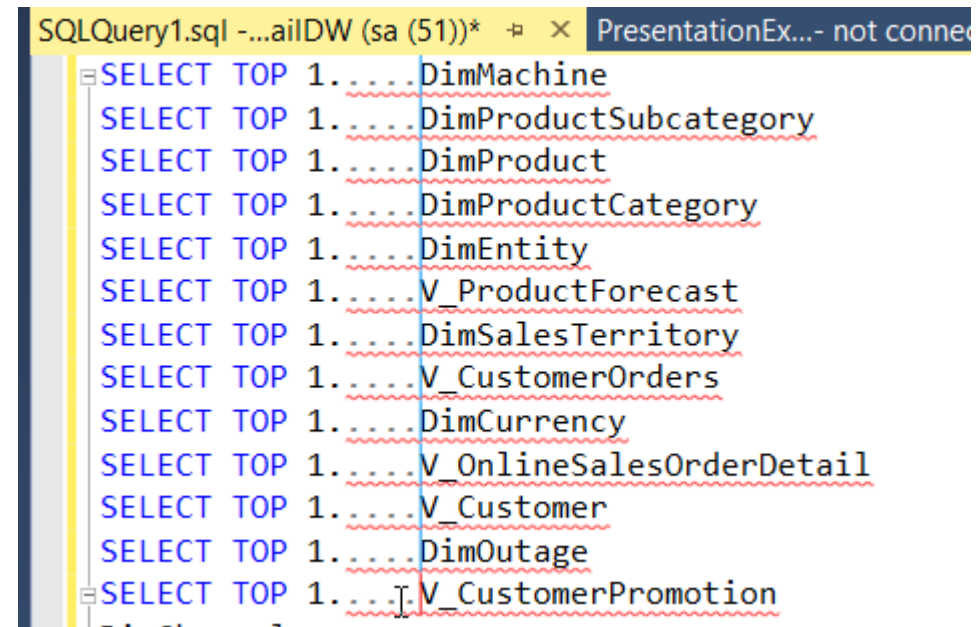


# SSMS Editing Multiple Lines Concurrently

- SSMS works pretty well  
Hold Alt + Shift + Arrow Up/Down to put a cursor on each line  
Now any typing will occur on all selected lines



```
SQLQuery1.sql -...ailDW (sa (51))* PresentationEx...- not connected*
DimMachine
DimProductSubcategory
DimProduct
DimProductCategory
DimEntity
V_ProductForecast
DimSalesTerritory
V_CustomerOrders
DimCurrency
V_OnlineSalesOrderDetail
V_Customer
DimOutage
V_CustomerPromotion
```



```
SQLQuery1.sql -...ailDW (sa (51))* PresentationEx...- not connected
SELECT TOP 1.....DimMachine
SELECT TOP 1.....DimProductSubcategory
SELECT TOP 1.....DimProduct
SELECT TOP 1.....DimProductCategory
SELECT TOP 1.....DimEntity
SELECT TOP 1.....V_ProductForecast
SELECT TOP 1.....DimSalesTerritory
SELECT TOP 1.....V_CustomerOrders
SELECT TOP 1.....DimCurrency
SELECT TOP 1.....V_OnlineSalesOrderDetail
SELECT TOP 1.....V_Customer
SELECT TOP 1.....DimOutage
SELECT TOP 1.....V_CustomerPromotion
```

# Visual Studio Code Editing Multiple Lines

---

- Visual Studio Code is even better...  
Hold Ctrl + Alt + Arrow Up/Down to put a cursor on each line  
Now any typing will occur on all selected lines just like in SSMS...
- So why's it better?
  - If you have multiple cursors on screen, you can do Ctrl + Arrow Right / Arrow Left, it will jump ALL the cursors to the end or beginning of the following word – SSMS will NOT do this

# WHERE 1=0 Magic or Top 0

---

-- Creates an empty shell copy of the DimCustomer table

```
SELECT *  
INTO dbo.DimCustomerCopy  
FROM dbo.DimCustomer  
WHERE 1=0
```

-- A meetup attendee also said you can do this...(it works too!)

```
SELECT TOP 0 *  
INTO dbo.DimCustomerCopy  
FROM dbo.DimCustomer
```

# Set Based Operations – See SQL file

---

UNION

UNION ALL

INTERSECT

EXCEPT

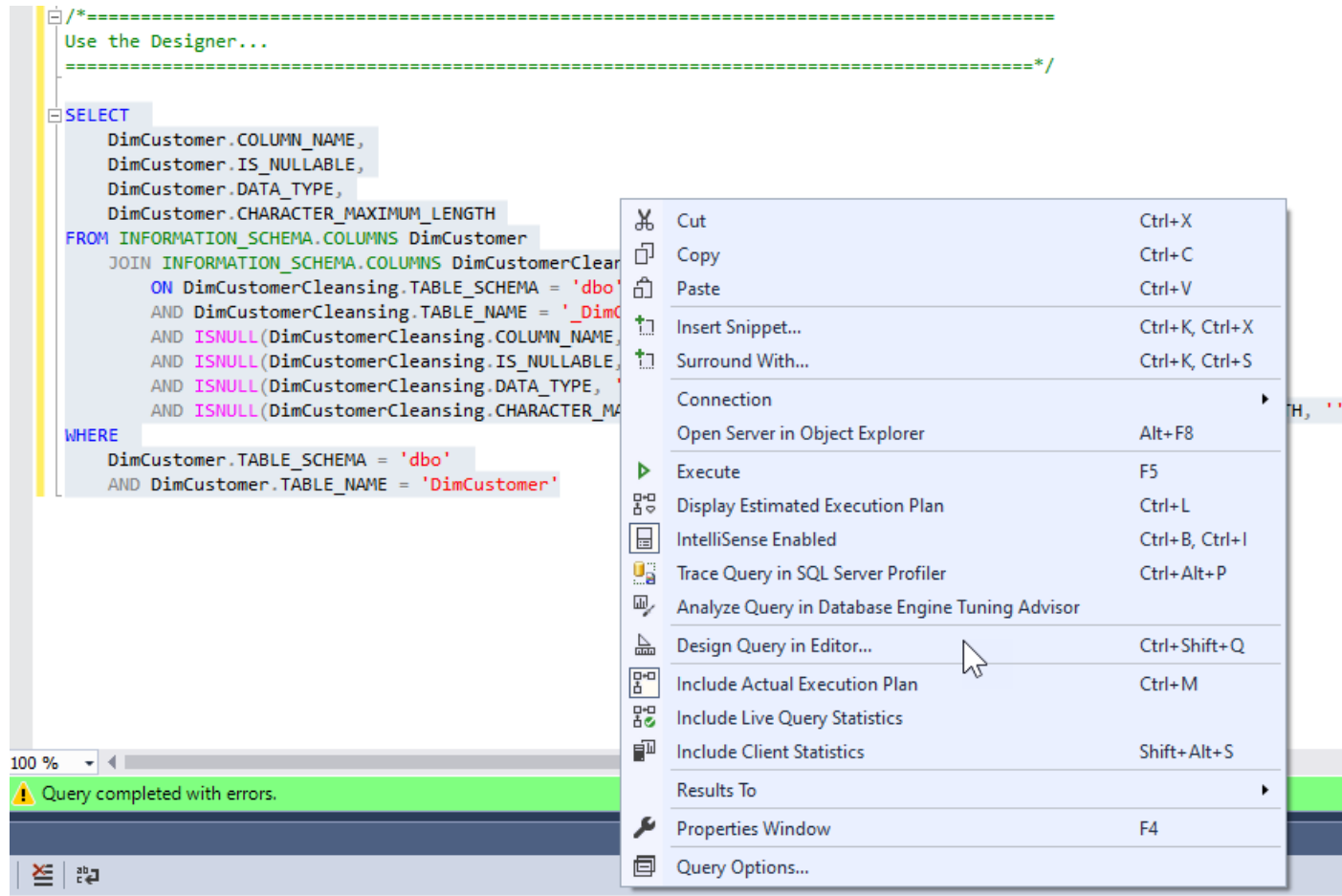


# ORDER BY Randomness

---

```
SELECT TOP 100 *  
FROM DimCustomer  
ORDER BY NEWID()
```

# Using the Query Editor – Still Useful



# Poor Man's T-SQL Formatter VSCode

---

All about it

- <http://architectshack.com/PoorMansTSqlFormatter.ashx>

Visual Studio Code

- <https://marketplace.visualstudio.com/items?itemName=TaoKlerks.poor-mans-t-sql-formatter-vscode>

# Azure Data Studio

<https://aka.ms/azuredatstudio>

For when you want to access  
SQL Server (or CosmosDB)  
Cross-Platform

- Windows
- Mac
- Linux

The screenshot displays the Azure Data Studio application window. The title bar indicates the active query is 'SQLQuery\_1 - ,1500:ContosoRetailDW (sa) - Azure Data Studio'. The interface is divided into several panes:

- EXPLORER:** Shows the file structure with 'Welcome', '1500:ContosoRetailDW', and 'SQLQuery\_1 - ,1500...DW (sa)'.
- Query Editor:** Contains the SQL query: `SELECT TOP 100 * FROM DimCustomer`. The status bar at the bottom of this pane shows 'Ln 2, Col 1 - Spaces: 4 - UTF-8 - CRLF - MSSQL - 100 rows - 00:00:00 - ,1500: ContosoRetailDW'.
- Results:** Displays a table with 15 columns: CustomerKey, GeographyKey, CustomerLabel, Title, FirstName, MiddleName, LastName, NameStyle, BirthDate, MaritalStatus, Suffix, Gender, EmailAddress, YearlyIncome, TotalChildren, NumberChildrenAtHome, and Education. The table contains 100 rows of data.
- Messages:** Currently empty.

The status bar at the bottom of the application window shows 'Ln 2, Col 1 - Spaces: 4 - UTF-8 - CRLF - MSSQL - 100 rows - 00:00:00 - ,1500: ContosoRetailDW'.