

```

/*****
Technology Enhancements for SQL Server 2014/2016 Developers - Demo file
By: Wylie Blanchard
Note: SQL Server 2016 is required for this demo
*****/

/**Memory-Optimized Tables**/

---- drop database InMemOLTP - if already exists
Use MASTER
DROP DATABASE if EXISTS InMemOLTP

-- Create Database Which Creates A File Group Containing Memory_Optimized_Data
---- create database
CREATE DATABASE InMemOLTP
ON PRIMARY(NAME = InMemOLTPData,
FILENAME = 'c:\data\InMemOLTPData.mdf', size=200MB),
---- memory optimized data
FILEGROUP [InMemOLTP_FG] CONTAINS MEMORY_OPTIMIZED_DATA(
NAME = [InMemOLTP_InMemOLTP_dir],
FILENAME = 'c:\data\InMemOLTP_InMemOLTP_dir')
LOG ON (name = [InMemOLTP_demo_log], filename='c:\data\InMemOLTP.ldf', size=100MB)
GO

-- Create A Regular Table & A Table With Setting Memory_Optimized Set To Enabled
USE InMemOLTP
GO
---- create a regular table
CREATE TABLE RegularTable (ID INT NOT NULL PRIMARY KEY,
Name VARCHAR(100) NOT NULL)
GO

---- create a memory optimized table
CREATE TABLE MemoryTable (ID INT NOT NULL,
Name VARCHAR(100) NOT NULL
CONSTRAINT ID_Clust_MemoryTable PRIMARY KEY NONCLUSTERED HASH (ID) WITH
(BUCKET_COUNT=1000000))
WITH (MEMORY_OPTIMIZED=ON)
GO

-- Create A Regular Stored Procedure - simple table to insert 100,000 rows
CREATE PROCEDURE Reglar_Insert_test
AS
BEGIN
SET NOCOUNT ON
DECLARE @counter AS INT = 1
DECLARE @start DATETIME
SELECT @start = GETDATE()
WHILE (@counter <= 100000)
BEGIN
INSERT INTO RegularTable VALUES(@counter, 'WylieBlanchard')
SET @counter = @counter + 1
END
SELECT DATEDIFF(SECOND, @start, GETDATE() ) [Regular_Insert in sec]
END
GO

```

```

-- Create A Natively Compiled Stored Procedure InMemOLTP table to insert 100,000 rows
CREATE PROCEDURE ImMemory_Insert_test
WITH NATIVE_COMPILATION, SCHEMABINDING, EXECUTE AS OWNER
AS
BEGIN ATOMIC WITH (TRANSACTION ISOLATION LEVEL=SNAPSHOT, LANGUAGE='english')
DECLARE @counter AS INT = 1
DECLARE @start DATETIME
SELECT @start = GETDATE()
WHILE (@counter <= 100000)
BEGIN
INSERT INTO dbo.MemoryTable VALUES(@counter, 'WylieBlanchard')
SET @counter = @counter + 1
END
SELECT DATEDIFF(SECOND, @start, GETDATE() ) [InMemOLTP_Insert in sec] END
GO

-- Compare the Performance of both Stored Procedures
---- insert data into [RegularTable] (100,000 rows)
EXEC Reglar_Insert_test
GO
---- insert data into [MemoryTable] (100,000 rows)
EXEC ImMemory_Insert_test
GO

/** T-SQL Statements**/

-- JSON format support
Select ID, Name FROM [dbo].[MemoryTable] FOR JSON AUTO
-- XML Example
Select ID, Name FROM [dbo].[MemoryTable] FOR XML AUTO

-- DROP IF EXISTS Statements
---- Old Drop Procedure Script **/
IF EXISTS (SELECT * FROM sys.procedures WHERE name = 'Reglar_Insert_test')
DROP PROCEDURE Reglar_Insert_test

---- New 2016 Drop Procedure Script
DROP PROCEDURE IF EXISTS [dbo].[ImMemory_Insert_test]

---- Old Drop Table Script
IF OBJECT_ID('[dbo].[RegularTable]', 'U') IS NOT NULL
DROP TABLE [dbo].[RegularTable];

---- New 2016 Drop Table Script
DROP TABLE IF EXISTS [dbo].[MemoryTable]

--TRUNCATE TABLE with PARTITION
TRUNCATE TABLE dbo.TableName WITH (PARTITIONS (1,5 TO 8))

/**Dynamic Data Masking (DDM)**/

```

```

-- drop database MaskingDatabase - if already exists
USE [master]
DROP DATABASE IF EXISTS [MaskingDatabase]

-- Create Database and table used for Masking Demo
---- create database
CREATE DATABASE MaskingDatabase
GO

---- Create Table with Different Data Type Columns
USE MaskingDatabase
CREATE TABLE AccountInfo (
ID INT IDENTITY(1, 1) PRIMARY KEY
,Name NVARCHAR(50) NOT NULL ,SSN VARCHAR(11) NULL
,Phone VARCHAR (15) NULL,Email NVARCHAR(60) NULL
,DocID INT NULL ,ICD10 NVARCHAR(7) NULL
,Date DATETIME NULL, Website VARCHAR (50)
)
GO

---- insert a row
INSERT INTO [dbo].[AccountInfo]
([Name],[SSN],[Phone],[Email],[DocID],[ICD10],[Date],[Website])
VALUES('Wylie Blanchard','123-45-6789','(312) 985-6810'
,'WBlanchard@greattechpros.com',1234,'A000000', GETDATE()
,'www.GreatTechPros.com' )
--,('Wylie Blanchard','987-65-4321','(312) 985-6810'
--, 'WylieBlanchard@gmail.com',9876,'Z999999', GETDATE()
--, 'www.WylieBlanchard.com' )
GO

---- review inserted row
SELECT * FROM AccountInfo
GO

-- Apply Masking to Table Columns
ALTER TABLE AccountInfo
ALTER COLUMN [SSN] ADD MASKED WITH (FUNCTION = 'partial(2,"XXX-XX-XXXX",2)')
ALTER TABLE AccountInfo
ALTER COLUMN [Email] ADD MASKED WITH (FUNCTION = 'email()')
ALTER TABLE AccountInfo
ALTER COLUMN [DocID] ADD MASKED WITH (FUNCTION = 'default()')-- default on int
ALTER TABLE AccountInfo
ALTER COLUMN [ICD10] ADD MASKED WITH (FUNCTION = 'default()')-- default on varchar
ALTER TABLE AccountInfo
ALTER COLUMN [Date] ADD MASKED WITH (FUNCTION = 'default()')-- default on date
GO

-- Create a New User and Grant Select Permissions
USE MaskingDatabase
GO
CREATE USER EndUser WITHOUT LOGIN;
GRANT SELECT ON AccountInfo TO EndUser;

-- Example of Selecting the data as user**/
---- select as admin user
SELECT * FROM AccountInfo; -- this would show clear data

```

```

GO
--select as user with read permission only
EXECUTE AS USER = 'EndUser';
SELECT * FROM AccountInfo -- this should show masked data
REVERT;
GO

/**Query Store**/

-- Enabling the Query Store for Existing Database
Use [TestDB]
ALTER DATABASE TestDB SET QUERY_STORE = ON;
/** Note:
Using the Query Store Page via Management Studio interface
In Object Explorer, right-click a database, and then click Properties.
In the Database Properties dialog box, select the Query Store page.
In the Operation Mode (Requested) box, select On.
**/

-- drop database qstore_demo - if already exists
USE MASTER
DROP DATABASE IF EXISTS qstore_demo

-- Create Demo Query Store Database
CREATE DATABASE [qstore_demo]
--ON PRIMARY
--( NAME = N'qs_demo', FILENAME = N'C:\qs_demo.mdf' , SIZE = 102400KB ,
--     MAXSIZE = 1024000KB , FILEGROWTH = 20480KB )
-- LOG ON
--( NAME = N'qs_demo_log', FILENAME = N'D:\C:\Program Files\Microsoft SQL
Server\MSSQL13.SQL2016\MSSQL\DATA\qs_demo_log.ldf' , SIZE = 20480KB ,
--     MAXSIZE = 1024000KB , FILEGROWTH = 20480KB )
ALTER DATABASE [qstore_demo] SET AUTO_UPDATE_STATISTICS OFF
ALTER DATABASE [qstore_demo] SET AUTO_CREATE_STATISTICS OFF
ALTER DATABASE [qstore_demo] SET RECOVERY SIMPLE
ALTER DATABASE [qstore_demo] SET QUERY_STORE = OFF

-- Create table sp and populate table

---- create a table
USE qstore_demo
GO
CREATE TABLE dbo.db_store (c1 CHAR(3) NOT NULL, c2 CHAR(3) NOT NULL, c3 SMALLINT NULL)
GO

---- create a stored procedure
CREATE PROC dbo.proc_1 @par1 SMALLINT
AS
SET NOCOUNT ON
SELECT c1, c2 FROM dbo.db_store
WHERE c3 = @par1
GO

---- populate the table (this may take a couple of minutes)
SET NOCOUNT ON

```

```

INSERT INTO [dbo].db_store (c1,c2,c3) SELECT '18','2f',2
go 20000
INSERT INTO [dbo].db_store (c1,c2) SELECT '171','1ff'
go 4000
INSERT INTO [dbo].db_store (c1,c2,c3) SELECT '172','1ff',0
go 10
INSERT INTO [dbo].db_store (c1,c2,c3) SELECT '172','1ff',4
go 15000

-- Enable Query Store on the database
ALTER DATABASE [qstore_demo] SET QUERY_STORE = ON
GO

-- *Test 1 - No Indexes on the Table
EXEC dbo.proc_1 0
GO 20

-- Review Query Store folder "Top Resource Consuming Queries"

-- *Test 2 - Testing with a Non Clustered Index*/
CREATE NONCLUSTERED INDEX NCI_1
ON dbo.db_store (c3)
GO
EXEC dbo.proc_1 0
GO 20
select * from db_store
-- Review Query Store "Top Resource Consuming Queries"

/** END **/
-- Clean Up below after demo
---- clean up -- drop database qstore_demo - if already exists
USE MASTER
DROP DATABASE IF EXISTS qstore_demo
---- clean up -- drop database MaskingDatabase - if already exists
USE MASTER
DROP DATABASE IF EXISTS MaskingDatabase
---- clean up -- drop database InMemOLTP - if already exists
Use MASTER
DROP DATABASE if EXISTS InMemOLTP

```