Technology Enhancements for SQL Server 2014/2016 Developers

Wylie Blanchard
Lead IT Consultant; SQL Server DBA

GTP GREAT TECH PROS

About Great Tech Pros

- Great Tech Pros was founded in 2012
- Specialties include:
  - IT Consulting
  - Database Administration, Management
  - Data Analysis
  - Website Design and Development
  - Professional Training and Presentations
- Visit us at www.GreatTechPros.com

GTP GREAT TECH PROS

Speaker: Wylie Blanchard

- SQL Server Database Consultant
- MCSE: SQL Server Data Platform
- MCSE: Data Management and Analytics
- Website: WylieBlanchard.com
- LinkedIn: in/WylieBlanchard
- Twitter: @WylieBlanchard1

- Pizza Connoisseur (self proclaimed)

Presentation Summary

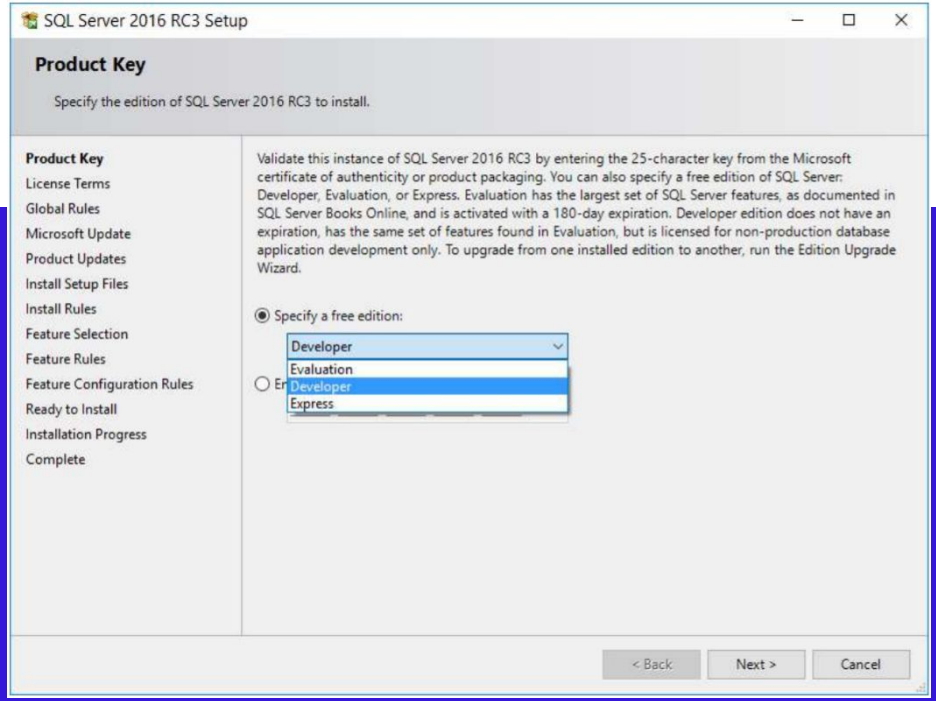
Learn what's new in SQL 2014/2016. Which features and enhancements are really important to the work life of a SQL Server Developer.

In this presentation we'll explore SQL Server 2014/2016 new possibilities, showing you how to use new T-SQL functions, features and enhancements that are only available in SQL Server 2014/2016.

What's New - SQL Server 2016

- Free Developer Edition
- Memory Optimized Tables
- Transact SQL
- Dynamic Data Masking
- Query Store

Free
Developer Edition



Free Developer Edition

- Full Feature set
- Sign in to (or sign-up for) Visual Studio Dev Essentials
- If you have a MSDN subscription you can download SQL Server 2016 Developer Edition from here.

Memory-Optimized Tables

In-Memory OLTP

- Memory Optimized Tables
  - Tables using the new data structures
- Allow highly used tables to live in memory
  - Remain in memory forever without losing records
- Designed to reduce blocking and locks
- Higher Performance response than disk tables due to data living in memory

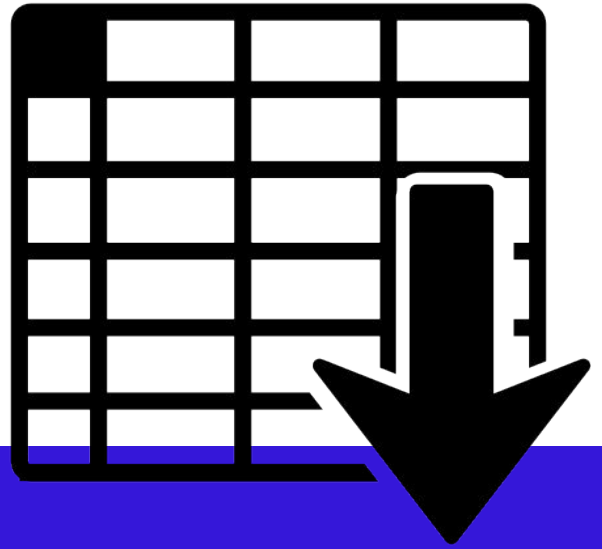
In-memory OLTP - Demo

Steps:

1. Create Database Which Creates A File Group Containing Memory_Optimized_Data
2. Create two different tables 1) Regular table and 2) Memory Optimized table
3. Create two stored procedures 1) Regular SP and 2) Natively Compiled SP
4. Compare the performance of two SPs

DEMO

Transact-SQL



JSON format support

- JSON: JavaScript Object Notation - is a syntax for storing and exchanging data
- JSON is an alternative to XML

- For more info about formatting Query Results as JSON with FOR JSON (SQL Server) click here

TRUNCATE TABLE w/ PARTITION

- TRUNCATE TABLE removes records and rests meta data for the entire table
- New … “TRUNCATE TABLE w/ PARTITION” allows remove records from specific partitions

Example:

```
TRUNCATE TABLE dbo.TableName WITH (PARTITIONS (1,5 TO 8))
```

DROP Statements

- Use DROP statements to remove existing entities.
  - Syntax has been enhanced for verifying whether the entity exists when dropping.

Syntax for SQL Server and Azure SQL Database

```
DROP TABLE [ IF EXISTS ] [ database_name . [
schema_name ] . | schema_name . ]
table_name [ ,...n ]
```

DROP IF EXISTS - Objects

- AGGREGATE
- PROCEDURE
- TABLE
- ASSEMBLY
- ROLE
- TRIGGER
- VIEW
- RULE
- DATABASE
- SCHEMA USERDEFAULT
- SECURITY POLICY
- VIEW
- FUNCTION
- SEQUENCE
- INDEX
- TYPE
- SYNONYM

DEMO

Dynamic Data Masking (DDM)

Dynamic Data Masking

- Limits sensitive data exposure by masking it to non-privileged users
- Helps prevent unauthorized access to sensitive data by enabling customers to designate how much of the sensitive data to reveal with minimal impact on the application layer.
- It’s a policy-based security feature that hides the sensitive data in the result set of a query over designated database fields, while the data in the database is not changed

DEMO

DDM Practices & Uses

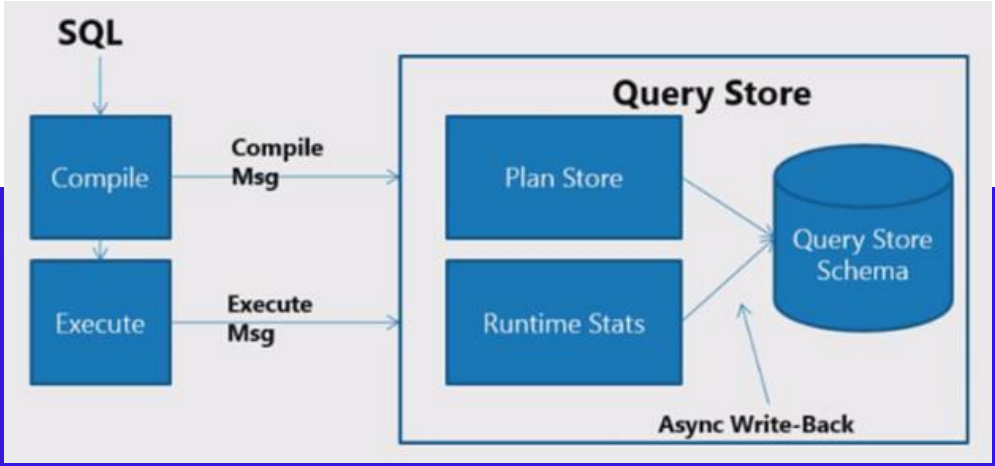
- Creating a mask on a column does not prevent updates to that column
- Using SELECT INTO or INSERT INTO to copy data from a masked column into another table results in masked data in the target table
- Will be applied when using SQL Server Import and Export.

DDM - Limitations

Masking rule can’t be used for the following column types:

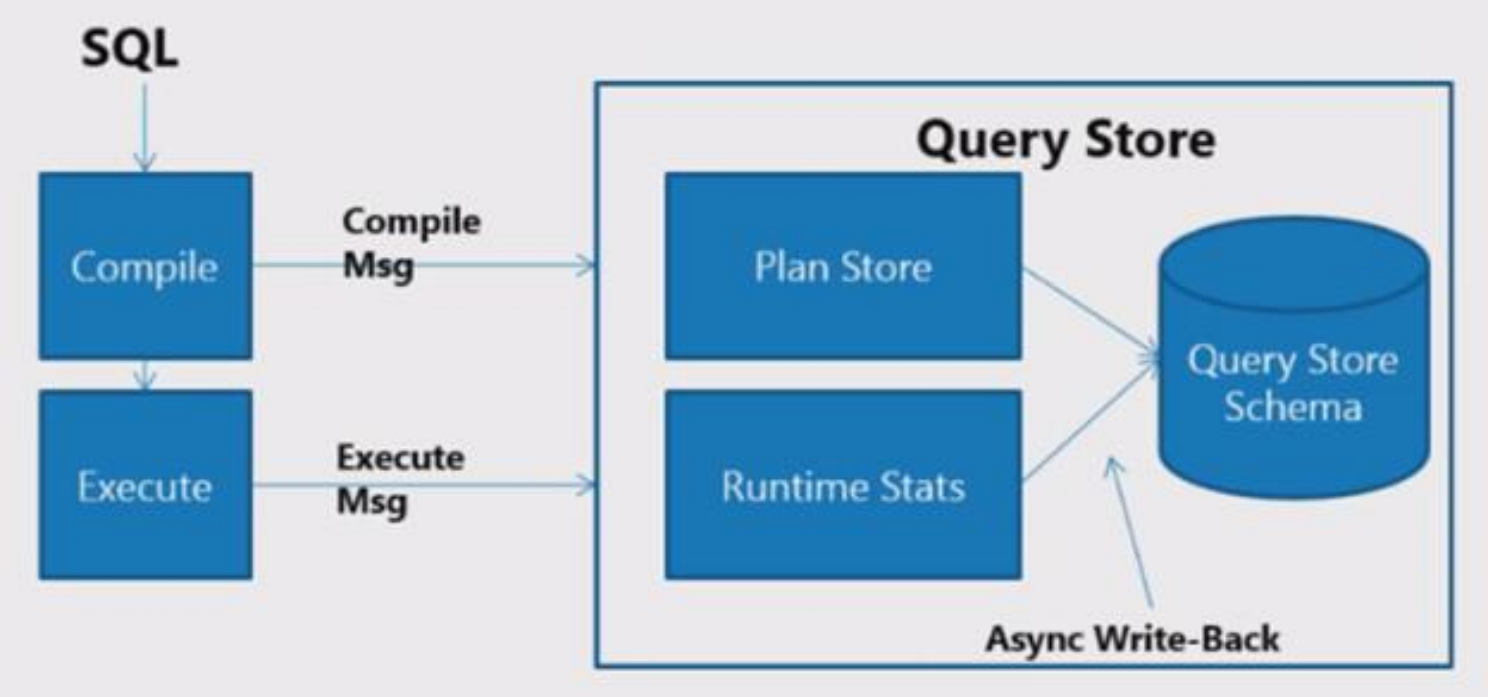
- Encrypted columns (Always Encrypted)
- FILESTREAM
- COLUMN_SET or a sparse column that is part of a column set.
- Can’t but used on computed column
  - but if the origin column(s) have a MASK, then the computed column will return masked data

Query Store

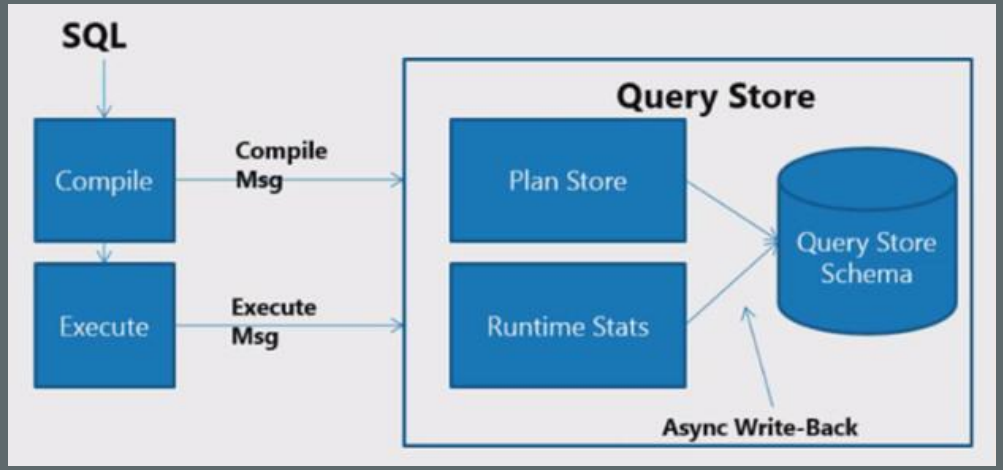

SQL
Compile
Execute
Compile Msg
Execute Msg
Query Store
Plan Store
Runtime Stats
Query Store Schema
Async Write-Back

Using the Query Store

- Helps simplify performance troubleshooting
- Capture queries, query plans, run time stats, etc.
- New system views created for Query Store
- View a history of Query workload


SQL
Compile
Execute
Compile
Msg
Execute
Msg
Query Store
Plan Store
Runtime Stats
Query Store
Schema
Async Write-Back

DEMO


SQL
Compile
Compile Msg
Execute
Execute Msg
Query Store
Plan Store
Runtime Stats
Query Store Schema
Async Write-Back

Other Features

Not mentioned in this presentation but are worth researching.

Live Query Statistics
Polybase
Advance Analytics
BI on Mobile Devices
Data Stretch to MS Azure
R Services

Resource Links

- In-Memory OLTP (In-Memory Optimization)
- Beginning In-Memory OLTP w/ Sample Example
- Format Query Results as JSON with FOR JSON
- DROP IF EXISTS – new thing in SQL Server 2016
- TRUNCATE TABLE w/ PARTITION
- Dynamic Data Masking
- SQL Server 2016 Query Store Example
- Monitoring Performance By Using the Query Store

Questions

Connect With Us

- Twitter: @GreatTechPros
- Linkedin: /company/Great-Tech-Pros
- Google+: +GreatTechPros
- Facebook: /GreatTechPros
- Website: GreatTechPros.com