Global Knowledge®

Expert Reference Series of White Papers

Are You Ready for Microsoft SQL Server 2016?

Are You Ready for Microsoft SQL Server 2016?

Brian D. Egler, Global Knowledge Course Director, MCITP, MCSE, MCT

Introduction

Are you ready to explore Microsoft SQL Server 2016? In this white paper we will examine three key new features that show how SQL Server 2016 provides automatic end-to-end security, seamless generation of business analytics, and elastic integration of data in the cloud.

Global Knowledge is the worldwide leader in IT and business training, delivering hands-on education via training facilities, private facilities, and the Internet, enabling our customers to choose when, where, and how they want to receive training programs and learning services. Keep an eye on the Global Knowledge website for details.

Microsoft SQL Server 2016 Themes

Three underlying themes categorize the SQL Server 2016 release:

- Mission Critical Performance
- Deeper Insights Across Data
- Hyperscale Cloud

Within this white paper, we will be exploring one selected new feature under each theme to get a sense of the capabilities of the new release. Other features, while too numerous to describe here, are documented on the Microsoft website. You can also download an evaluation copy of the software from the same location. Of course, all the functionality of Microsoft SQL Server 2016 will be contained in the Microsoft Official Curriculum (MOC) courses offered by Global Knowledge both in the physical and virtual classroom platforms when available.

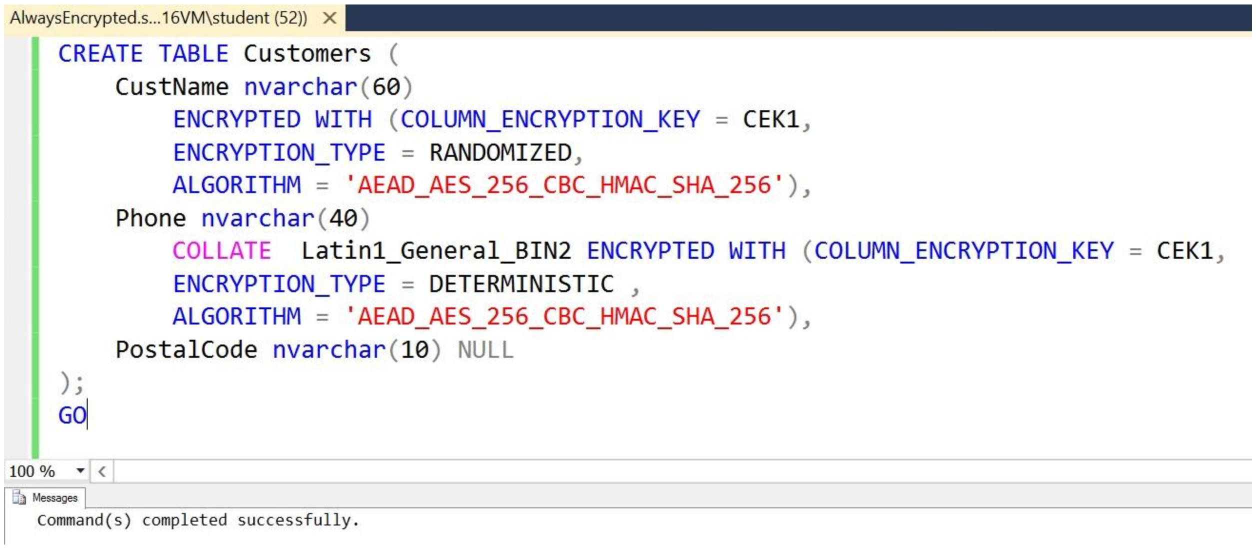
Mission Critical Performance

Selected Feature: *Always Encrypted*

Starting with SQL Server 2005, Microsoft allowed column-level encryption natively within the database engine. Sensitive columns could be encrypted by an application and decrypted as needed providing an “end-to-end” security option. Selected columns would be encrypted on disk, on backup, in memory, and over the network. However, column-level encryption required application code changes to use functions such as EncryptByKey and DecryptByKey. A Database Encryption Key and an appropriate Certificate were needed to be accessible for encryption or decryption to occur successfully. With SQL Server 2008, Microsoft introduced Transparent Data Encryption (TDE) as a feature that would automatically encrypt a whole database without having to change application code even for third-party applications. A major benefit was that the “data at rest” would be encrypted, including database files on disk and backups on tape or disk. This feature provided additional security to counter possible physical security vulnerabilities. However, as soon as data was “in motion,” for instance, into memory or over the network, the information was automatically decrypted and therefore not protected.

Now with SQL Server 2016, Microsoft has implemented an "Always Encrypted" option which includes end-to-end encryption without the need for application code changes therefore providing a best-of-both-worlds solution. A major benefit of this strategy is that Always Encrypted provides a separation between users who own the data and users who manage the data, such as administrators, because the encryption/decryption occurs at the client layer. With the advent of cloud-based data, this separation is especially important. A Column Master Key (CMK) and Column Encryption Key (CEK) are required to be accessible for encryption or decryption to occur successfully using an Always Encrypted-enabled driver installed on the client computer.

Although the application code does not need to change, the data definitions for sensitive columns will need to be redefined to include the ENCRYPTED WITH clause (see Figure 1).



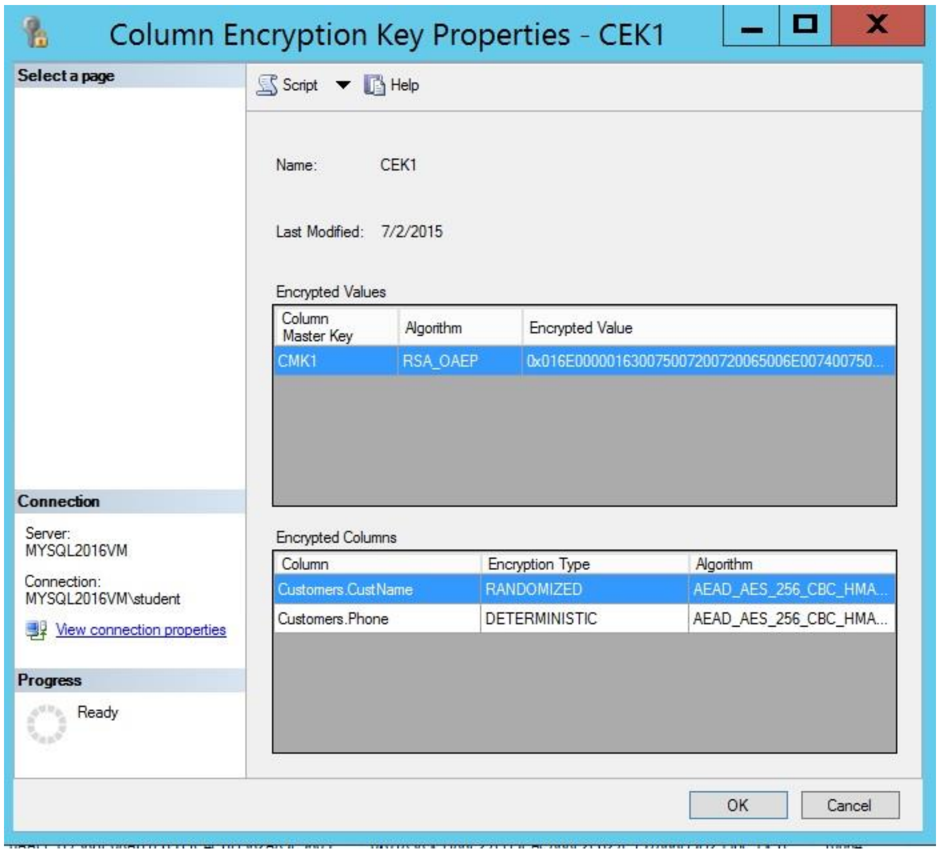
```
CREATE TABLE Customers (
    CustName nvarchar(60)
        ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = CEK1,
        ENCRYPTION_TYPE = RANDOMIZED,
        ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256'),
    Phone nvarchar(40)
        COLLATE  Latin1_General_BIN2 ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = CEK1,
        ENCRYPTION_TYPE = DETERMINISTIC ,
        ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256'),
    PostalCode nvarchar(10) NULL
);
GO
```

```
Command(s) completed successfully.
```

Figure 1: CREATE TABLE Example for "Always Encrypted"

The ENCRYPTION_TYPE clause for a column can be DETERMINISTIC or RANDOMIZED. The DETERMINISTIC setting means that, given a certain input data, the encrypted data will always be the same output. This option allows grouping, filtering, joining, and indexing with encrypted values. However, it may provide an opportunity for unauthorized users to deduce data for columns with few distinct values. The RANDOMIZED setting provides more security by encrypting in a less predictable way; however, it does not provide support for grouping, filtering, joining, and indexing with encrypted values.

Two types of encryption keys are required for encryption: A Column Encryption Key (CEK) and a Column Master Key (CMK). The CEK is used to perform fast, synchronous data encryption while the CMK is used to encrypt the CEK asynchronously for high security. The CMK must be available in a Trusted Key Store, typically using a Certificate, on the client machine. See Figure 2 for an example of Column Encryption Key Properties within SQL Server Management Studio (SSMS).



Name: CEK1

Last Modified: 7/2/2015

Encrypted Values

Column Master Key	Algorithm	Encrypted Value
CMK1	RSA_OAEP	0x016E000001630075007200720065006E007400750...

Encrypted Columns

Column	Encryption Type	Algorithm
Customers.CustName	RANDOMIZED	AEAD_AES_256_CBC_HMA...
Customers.Phone	DETERMINISTIC	AEAD_AES_256_CBC_HMA...

Server:
MYSQL2016VM

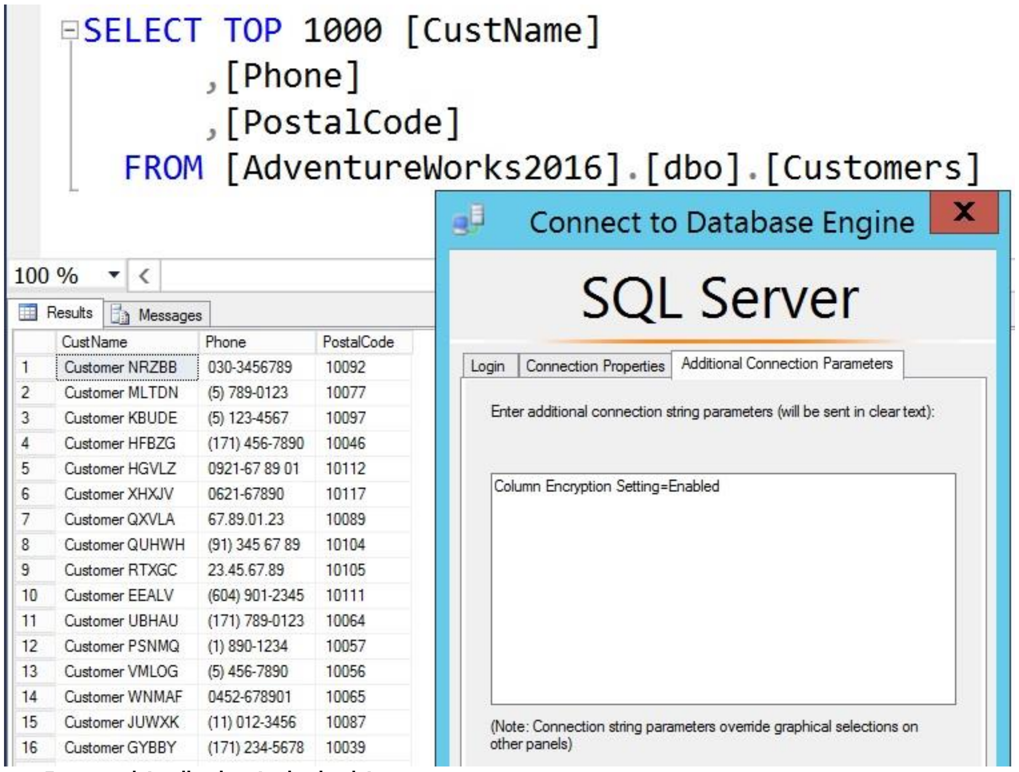
Connection:
MYSQL2016VM\student

View connection properties

Ready

Figure 2: Column Encryption Key Properties

Once the appropriate keys and certificates are available and authorized to a client, a special connection string option needs to be specified, namely, "Column Encryption Setting=Enabled" (see Figure 3).



```
SELECT TOP 1000 [CustName]
      ,[Phone]
      ,[PostalCode]
  FROM [AdventureWorks2016].[dbo].[Customers]
```

	CustName	Phone	PostalCode
1	Customer NRZBB	030-3456789	10092
2	Customer MLTDN	(5) 789-0123	10077
3	Customer KBUDE	(5) 123-4567	10097
4	Customer HFBZG	(171) 456-7890	10046
5	Customer HGVLZ	0921-67 89 01	10112
6	Customer XHXJV	0621-67890	10117
7	Customer QXVLA	67.89.01.23	10089
8	Customer QUHWH	(91) 345 67 89	10104
9	Customer RTXGC	23.45.67.89	10105
10	Customer EEALV	(604) 901-2345	10111
11	Customer UBHAU	(171) 789-0123	10064
12	Customer PSNMQ	(1) 890-1234	10057
13	Customer VMLOG	(5) 456-7890	10056
14	Customer WNMAF	0452-678901	10065
15	Customer JUWXK	(11) 012-3456	10087
16	Customer GYBBY	(171) 234-5678	10039

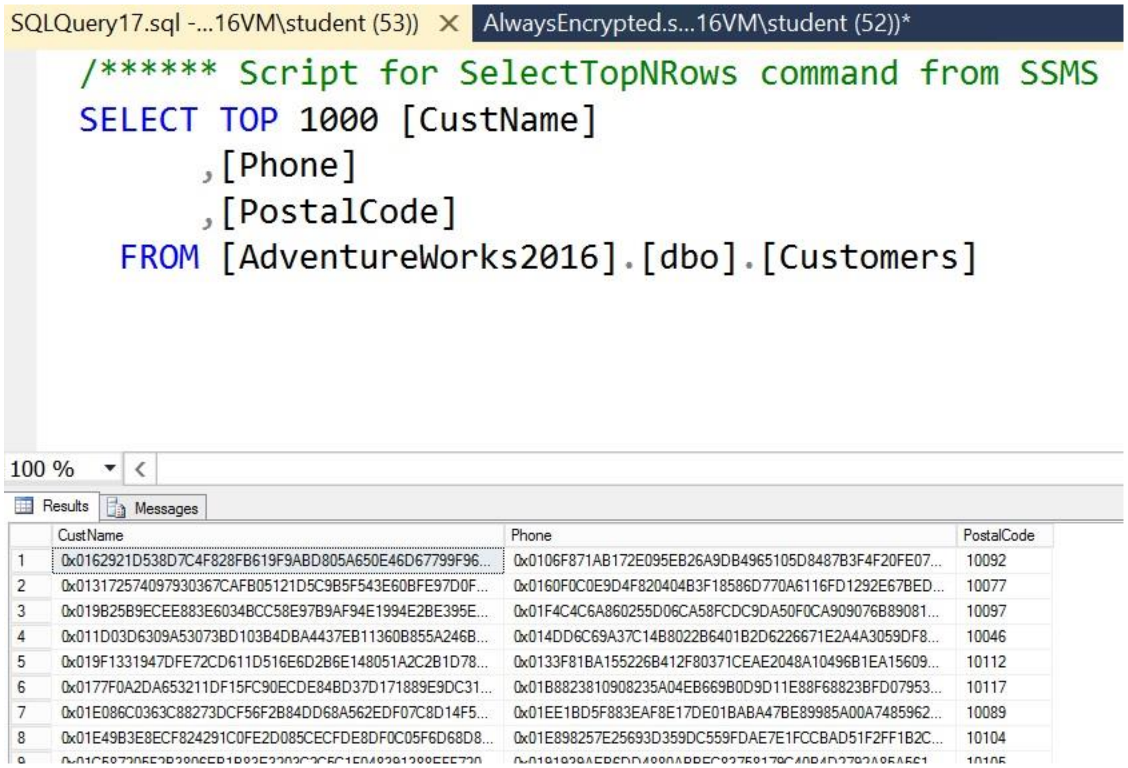
Enter additional connection string parameters (will be sent in clear text):

Column Encryption Setting=Enabled

(Note: Connection string parameters override graphical selections on other panels)

Figure 3: Always Encrypted Application Authorized Access

If any of the required objects are inaccessible or unauthorized, decryption will not occur and data values will be presented as encrypted, assuming the user has security permissions on the data columns themselves (see Figure 4) for an example where decryption is unauthorized but a user, such as an administrator, has access permissions.


SQLQuery17.sql -...16VM\student (53))
AlwaysEncrypted.s...16VM\student (52))*
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS
SELECT TOP 1000 [CustName]
      ,[Phone]
      ,[PostalCode]
  FROM [AdventureWorks2016].[dbo].[Customers]
```

	CustName	Phone	PostalCode
1	0x0162921D538D7C4F828FB619F9ABD805A650E46D67799F96...	0x0106F871AB172E095EB26A9DB4965105D8487B3F4F20FE07...	10092
2	0x013172574097930367CAFB05121D5C9B5F543E60BFE97D0F...	0x0160F0C0E9D4F820404B3F18586D770A6116FD1292E67BED...	10077
3	0x019B25B9ECEE883E6034BCC58E97B9AF94E1994E2BE395E...	0x01F4C4C6A860255D06CA58FCDC9DA50F0CA909076B89081...	10097
4	0x011D03D6309A53073BD103B4DBA4437EB11360B855A246B...	0x014DD6C69A37C14B8022B6401B2D6226671E2A4A3059DF8...	10046
5	0x019F1331947DFE72CD611D516E6D2B6E148051A2C2B1D78...	0x0133F81BA155226B412F80371CEAE2048A10496B1EA15609...	10112
6	0x0177F0A2DA653211DF15FC90ECDE84BD37D171889E9DC31...	0x01B8823810908235A04EB669B0D9D11E88F68823BFD07953...	10117
7	0x01E086C0363C88273DCF56F2B84DD68A562EDF07C8D14F5...	0x01EE1BD5F883EAF8E17DE01BABA47BE89985A00A7485962...	10089
8	0x01E49B3E8ECF824291C0FE2D085CECFDE8DF0C05F6D68D8...	0x01E898257E25693D359DC559FDAE7E1FCCBAD51F2FF1B2C...	10104
9	[illegible]	[illegible]	10105

Figure 4: Example Data Access without Decryption Authorization

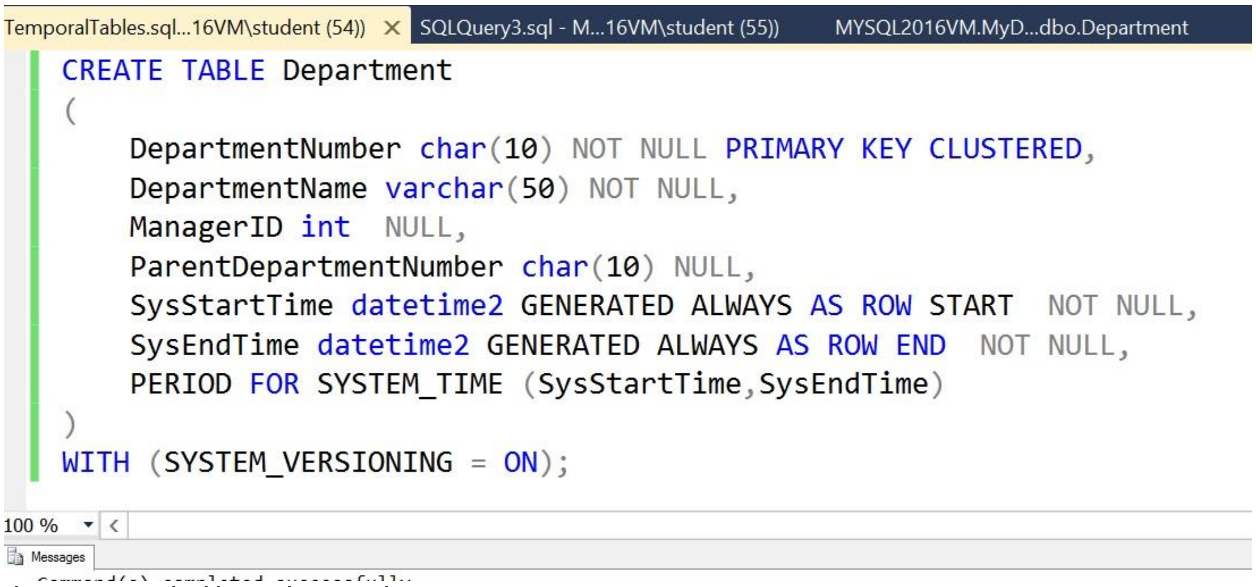
Always Encrypted is therefore a mission-critical feature of Microsoft SQL Server 2016 for both "on-premises" and "cloud-based" data that require an automatic and transparent end-to-end security solution.

Deeper Insights Across Data

Selected Feature: *Temporal Data*

When comparing the differences between operational systems and business intelligence (BI) solutions, we typically see that the operational viewpoint is that of the current, latest data values whereas the importance of BI provides the ability to analyze historical trends and patterns of data changes over time. A common way of supporting historical analysis in a BI solution is by defining a Slowly Changing Dimension (SCD) in the Data Warehouse (DW) design. The DW provides a historical database as the foundation for business analysis supporting Facts and Dimensions in a "Dimensional Model." Facts are typically numeric values that are used to assess the business, such as revenue, costs, profits, salaries, etc. Dimensions add context to the analysis of facts, for instance, analyzing revenue by region or profits by year or average manager salary by department. In these examples, region, year, and department would be dimensions in the DW with revenue, profits, and salary implemented as facts. Consider when a new manager is assigned to a department, should their salary be applied to the whole year average or only the time period that they were assigned? If we choose the former, we are defining an "SCD Type 1," which only stores current data values and the potential for inaccurate detailed analysis. For accurate analysis over time periods, we must choose the latter option, which requires the careful design of an "SCD Type 2" for historical data. The typical design method of an SCD Type 2 is to provide the ability for multiple rows for a given dimension member to allow for storage of historical changes by the addition of a "surrogate" key. The surrogate key provides uniqueness and allows duplicate values of the business key used in the operational system. Typically added also, would be a start date/time and an end date/time column for accurate time analysis. The design and population of this type of dimension is usually a complex, manual, time-consuming process. Now in SQL Server 2016, we have the option of Temporal Data, which automates this process via system tables and is part of the ANSI SQL 2011 specification.

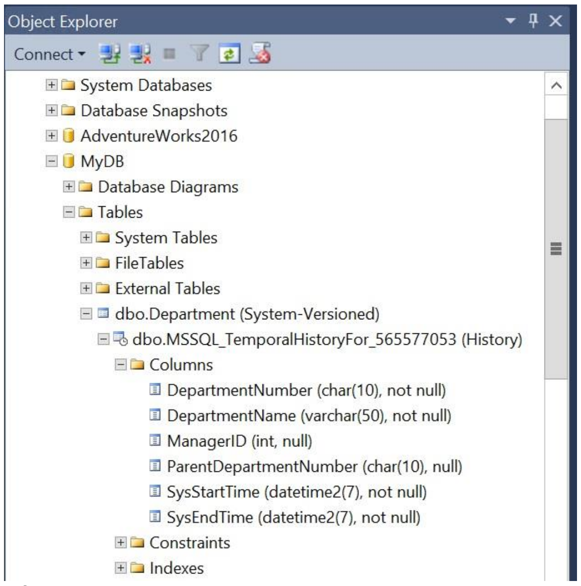
Temporal Tables, also known as "system-versioned tables," can be defined in any database and will automatically track data changes with start and end date/times. The temporal table actually consists of two tables, the current table that shows the latest data values and a history table, which shows the previous changes and when they were current. See Figure 5 for an example of a temporal table definition.


TemporalTables.sql...16VM\student (54))
SQLQuery3.sql - M...16VM\student (55))
MYSQL2016VM.MyD...dbo.Department
100 %
Messages

```
CREATE TABLE Department
(
    DepartmentNumber char(10) NOT NULL PRIMARY KEY CLUSTERED,
    DepartmentName varchar(50) NOT NULL,
    ManagerID int  NULL,
    ParentDepartmentNumber char(10) NULL,
    SysStartTime datetime2 GENERATED ALWAYS AS ROW START  NOT NULL,
    SysEndTime datetime2 GENERATED ALWAYS AS ROW END  NOT NULL,
    PERIOD FOR SYSTEM_TIME (SysStartTime,SysEndTime)
)
WITH (SYSTEM_VERSIONING = ON);
```

Figure 5: Temporal Table Creation Example

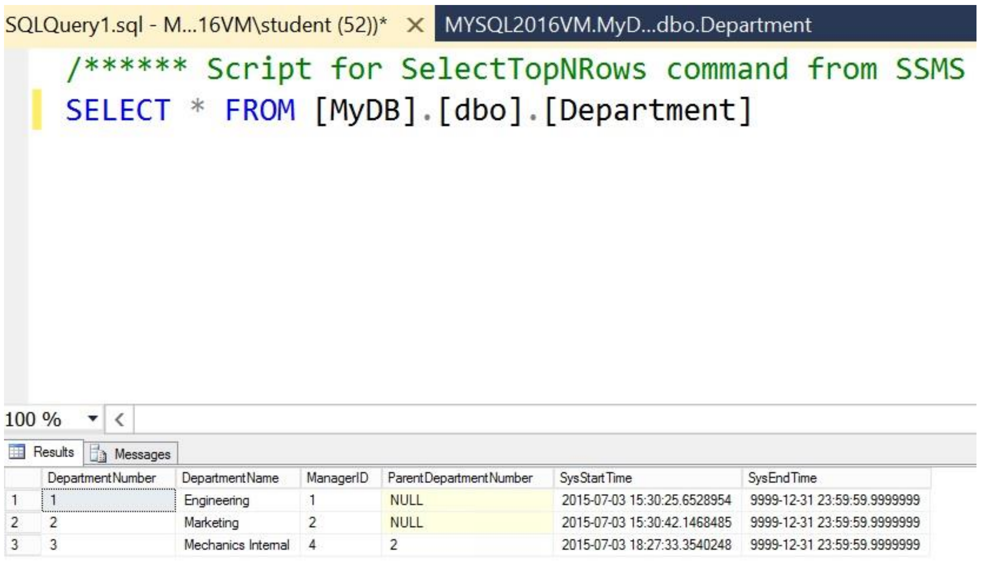
Once created, the temporal table will automatically track changes and maintain the start and end date/time columns, effectively implementing an SCD Type 2 via the history table while keeping the current table as normal with the latest data values. Both tables are visible in the SSMS Object Explorer (Figure 6) and can be queried as separate tables, as required.



```
⊞ System Databases
⊞ Database Snapshots
⊞ AdventureWorks2016
⊟ MyDB
    ⊞ Database Diagrams
    ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊟ dbo.Department (System-Versioned)
            ⊟ dbo.MSSQL_TemporalHistoryFor_565577053 (History)
                ⊟ Columns
                    DepartmentNumber (char(10), not null)
                    DepartmentName (varchar(50), not null)
                    ManagerID (int, null)
                    ParentDepartmentNumber (char(10), null)
                    SysStartTime (datetime2(7), not null)
                    SysEndTime (datetime2(7), not null)
                ⊞ Constraints
                ⊞ Indexes
```

Figure 6: Temporal Table in Object Explorer

Notice that the current table represents the latest data only with the SysEndTime column set to the maximum value for datatype datetime2 to indicate current values. See Figure 7.

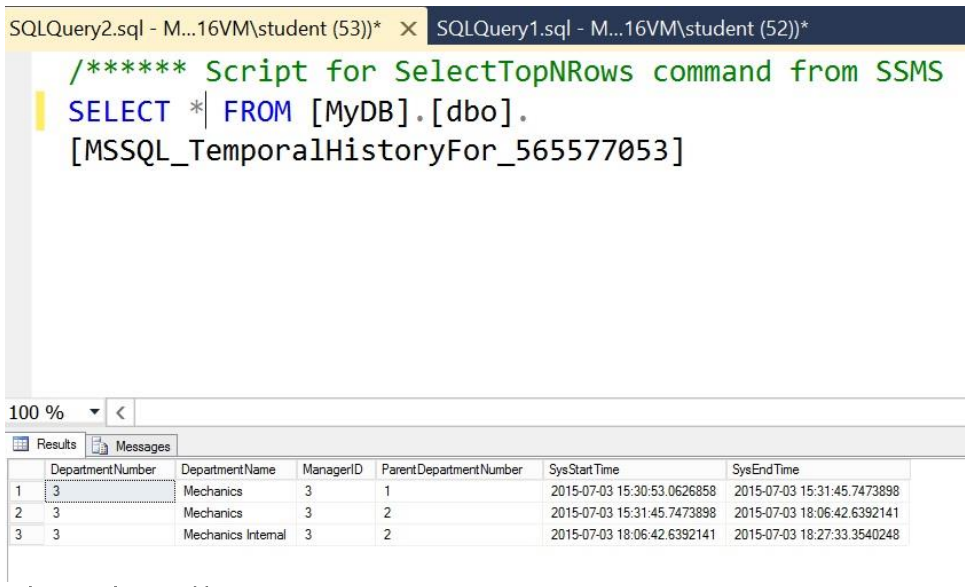

SQLQuery1.sql - M...16VM\student (52))*
MYSQL2016VM.MyD...dbo.Department
100 %
Results
Messages

```sql
/****** Script for SelectTopNRows command from SSMS
SELECT * FROM [MyDB].[dbo].[Department]
```

	DepartmentNumber	DepartmentName	ManagerID	ParentDepartmentNumber	SysStartTime	SysEndTime
1	1	Engineering	1	NULL	2015-07-03 15:30:25.6528954	9999-12-31 23:59:59.9999999
2	2	Marketing	2	NULL	2015-07-03 15:30:42.1468485	9999-12-31 23:59:59.9999999
3	3	Mechanics Internal	4	2	2015-07-03 18:27:33.3540248	9999-12-31 23:59:59.9999999

Figure 7: Temporal Data - Current Table Query

Also notice that that the history table represents the previous updates with the SysStartTime and SysEndTime columns representing the period of time that those data values were current. Figure 8 shows that the DepartmentNumber “3” row has been updated three times after initial insertion. Temporal data can be used for auditing or for recovering data values after accidental updates but most importantly can be directly copied to an SCD Type 2 dimension table in a Data Warehouse without manipulation, saving complex processing when incrementally loading data.



SQLQuery2.sql - M...16VM\student (53))*
SQLQuery1.sql - M...16VM\student (52))*
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS
SELECT * FROM [MyDB].[dbo].
[MSSQL_TemporalHistoryFor_565577053]
```

	DepartmentNumber	DepartmentName	ManagerID	ParentDepartmentNumber	SysStartTime	SysEndTime
1	3	Mechanics	3	1	2015-07-03 15:30:53.0626858	2015-07-03 15:31:45.7473898
2	3	Mechanics	3	2	2015-07-03 15:31:45.7473898	2015-07-03 18:06:42.6392141
3	3	Mechanics Internal	3	2	2015-07-03 18:06:42.6392141	2015-07-03 18:27:33.3540248

Figure 8: Temporal Data - History Table Query

Hyperscale Cloud

Selected Feature: *Stretch Database*

Microsoft's transition to a cloud-computing, services-based company has been rapid and largely successful with major services such as Office 365 and Azure leading the way. Cloud-based databases provide increased flexibility and "elastic" scalability, ideal for rapidly growing organizations. There are still many evolving security-based questions, but some data may be appropriate to host in the cloud and synchronize with sensitive "on-premises" data. This distributed approach to our data is a growing trend in the industry. Microsoft is at the forefront of this technology with its Azure platform supporting Azure SQL Database in the cloud and also Azure Virtual Machines (VMs), which can support the full version of SQL Server, if needed. Once again, ease of use and lower costs are Microsoft's value propositions. Azure SQL Database is now compatible with SQL Server 2016 and supports a new hybrid option called the "Stretch Database."

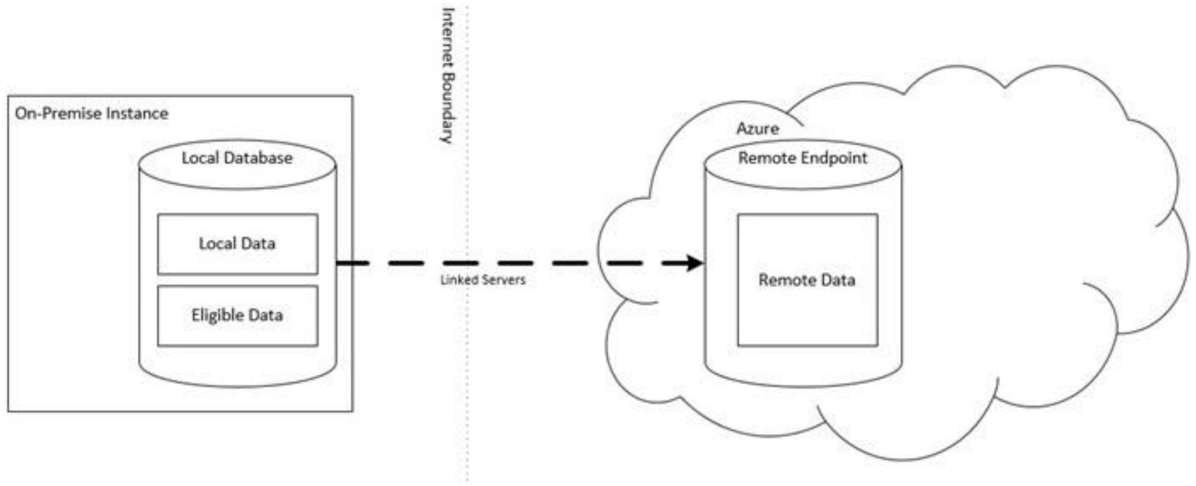


Figure 9: Stretch Database Concept

The Stretch Database concept is designed for a database that has large historical tables that are required for query access but in an infrequent manner. The objective is to store the most frequently accessed tables locally with the occasionally accessed historical tables off-loaded to the cloud, reducing local storage costs and resource demands. Application queries require no changes providing a transparent implementation.

The Stretch Database consists of a local database hosted by an on-premises instance of SQL Server 2016 and a remote database hosted by Azure SQL Database (V12 and above). Large historical tables can be identified as "stretch tables" making the data "eligible" for migration to the remote database. The stretch tables are fully migrated over time to the remote database, however, migration can be paused and resumed as required to optimize available network resources. Fail-safe mechanisms are built in so that no data loss will occur during migration including automatic retry functionality. Again, existing queries can remain unchanged both during and after migration as the query optimizer understands the current state of migration. Effectively, a stretch table consists of a local table with eligible data and a remote table with migrated data. After full migration, the local table will be empty and the remote table will contain all the data. Any new rows added to the local table will then become eligible for migration. A stretch-enabled table does not support update or delete operations.

In order to allow a stretch database to be hosted by a SQL Server 2016 instance, the "remote data archive" option must be turned on by using sp_configure. Then the required database can be enabled for stretch operations using the appropriate task wizard in SSMS. See Figure 10.

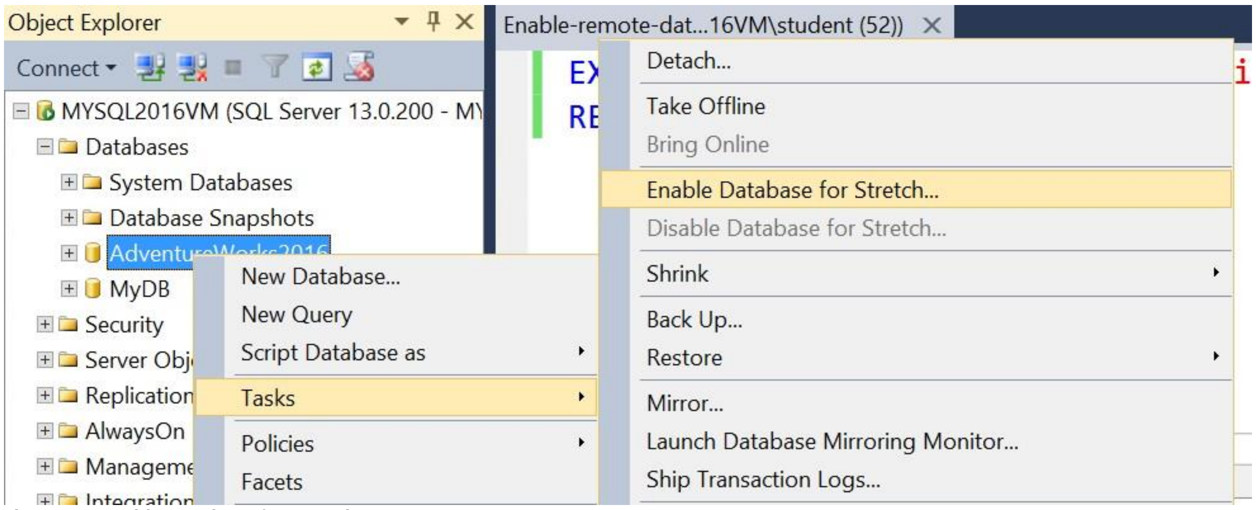
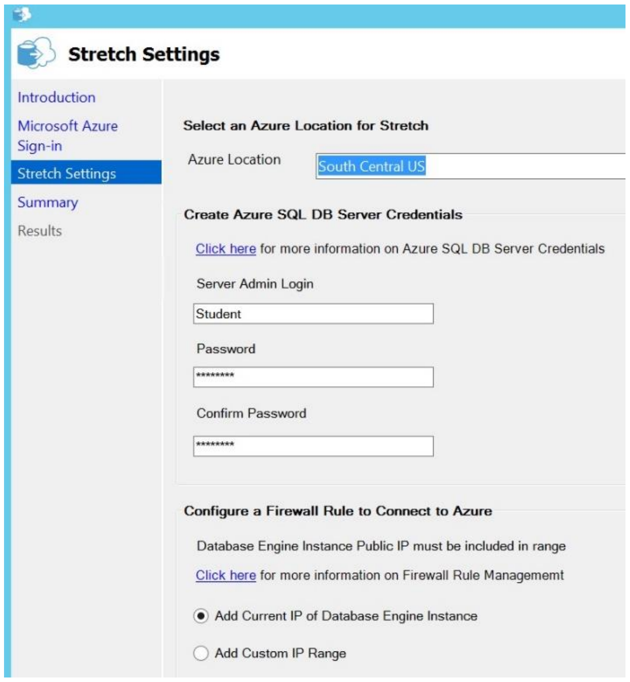


Figure 10: Enable Database for Stretch

Once the Enable Database for Stretch wizard starts, it prompts for the Azure subscription and sign-in credentials that should be used for the remote database server. The wizard creates a linked server definition to the remote Azure SQL Database for secure access. See Figure 11.



Select an Azure Location for Stretch

Azure Location: South Central US

Create Azure SQL DB Server Credentials

Click here for more information on Azure SQL DB Server Credentials

Server Admin Login: Student

Password: ********

Confirm Password: ********

Configure a Firewall Rule to Connect to Azure

Database Engine Instance Public IP must be included in range

Click here for more information on Firewall Rule Managememt

(●) Add Current IP of Database Engine Instance

( ) Add Custom IP Range

Figure 11: Stretch Settings

When the database is enabled for stretch operations, you can enable selected stretch tables within the database using the Enable Stretch option in SSMS. Again, occasionally accessed historical data tables would be ideal candidates for stretch tables. See Figure 12.

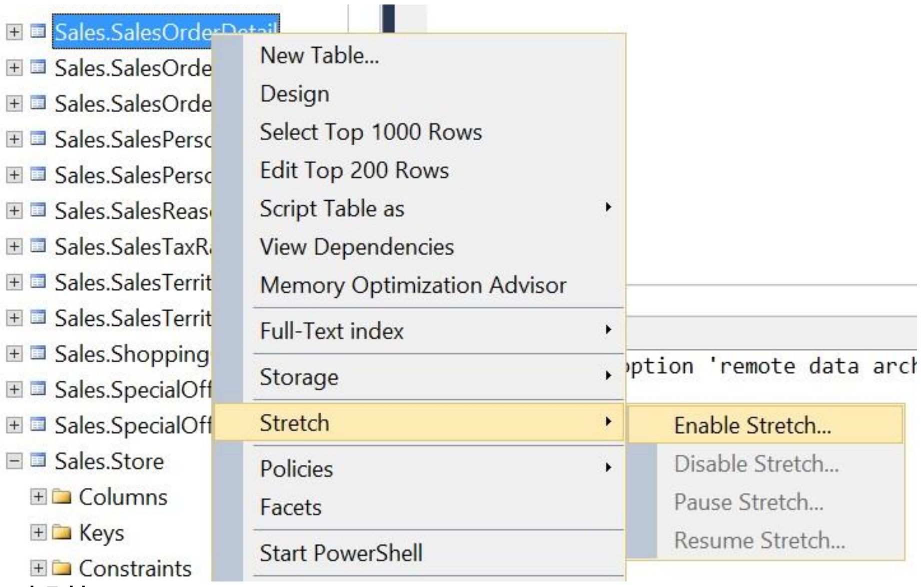
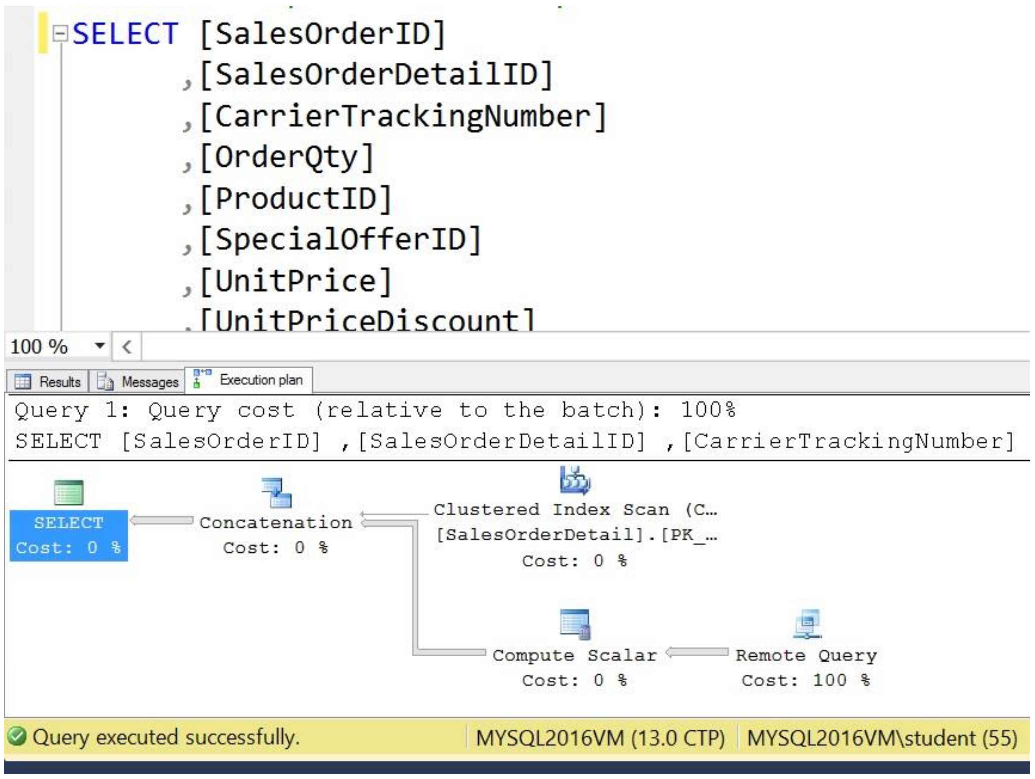


Figure 12: Enable Stretch Table

Once enabled, the stretch table will begin migration of data from the local table store to the remote table store. The migration can be paused and resumed as necessary to control network usage. Application queries against the stretch-enabled table remain unchanged, however, the query optimizer will create an execution plan that includes access to the local table and the remote table combined. (See Figure 13) Therefore regardless of the current state of migration, the correct data will be returned by the query. Over time, the entire table data will be migrated to the remote table-store freeing up storage and resources for the more active local, non-stretch tables. The stretch table will be available for occasional access as needed however, it must be noted that performance of remote queries against stretch tables will be slower than the access to local high-performance non-stretch tables, by design. As always, performance metrics must be assessed for acceptability to established service-level agreements.

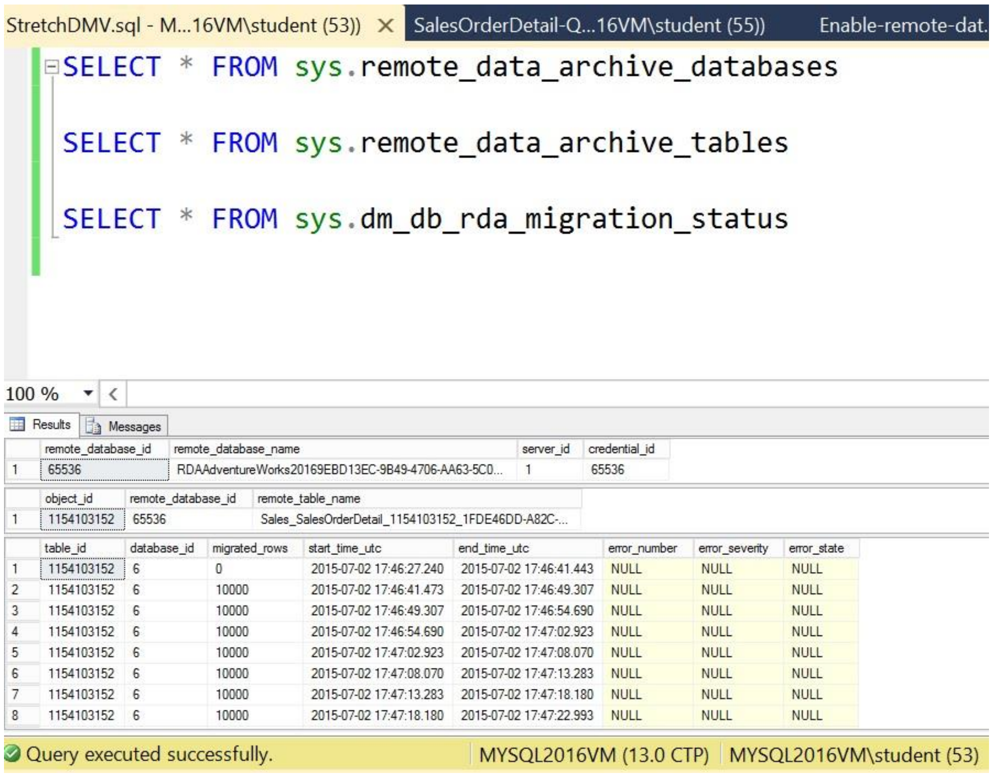


```
SELECT [SalesOrderID]
      ,[SalesOrderDetailID]
      ,[CarrierTrackingNumber]
      ,[OrderQty]
      ,[ProductID]
      ,[SpecialOfferID]
      ,[UnitPrice]
      ,[UnitPriceDiscount]
```

```
Query 1: Query cost (relative to the batch): 100%
SELECT [SalesOrderID] ,[SalesOrderDetailID] ,[CarrierTrackingNumber]
```

Figure 13: Query Plan for a Stretch Table

You can interrogate the system catalog for information regarding stretch databases and tables using new system catalog views and dynamic management objects as shown in Figure 14.


StretchDMV.sql - M...16VM\student (53)) × SalesOrderDetail-Q...16VM\student (55)) Enable-remote-dat.

```
SELECT * FROM sys.remote_data_archive_databases

SELECT * FROM sys.remote_data_archive_tables

SELECT * FROM sys.dm_db_rda_migration_status
```

100 %

Results | Messages

	remote_database_id	remote_database_name	server_id	credential_id
1	65536	RDAAdventureWorks20169EBD13EC-9B49-4706-AA63-5C0...	1	65536

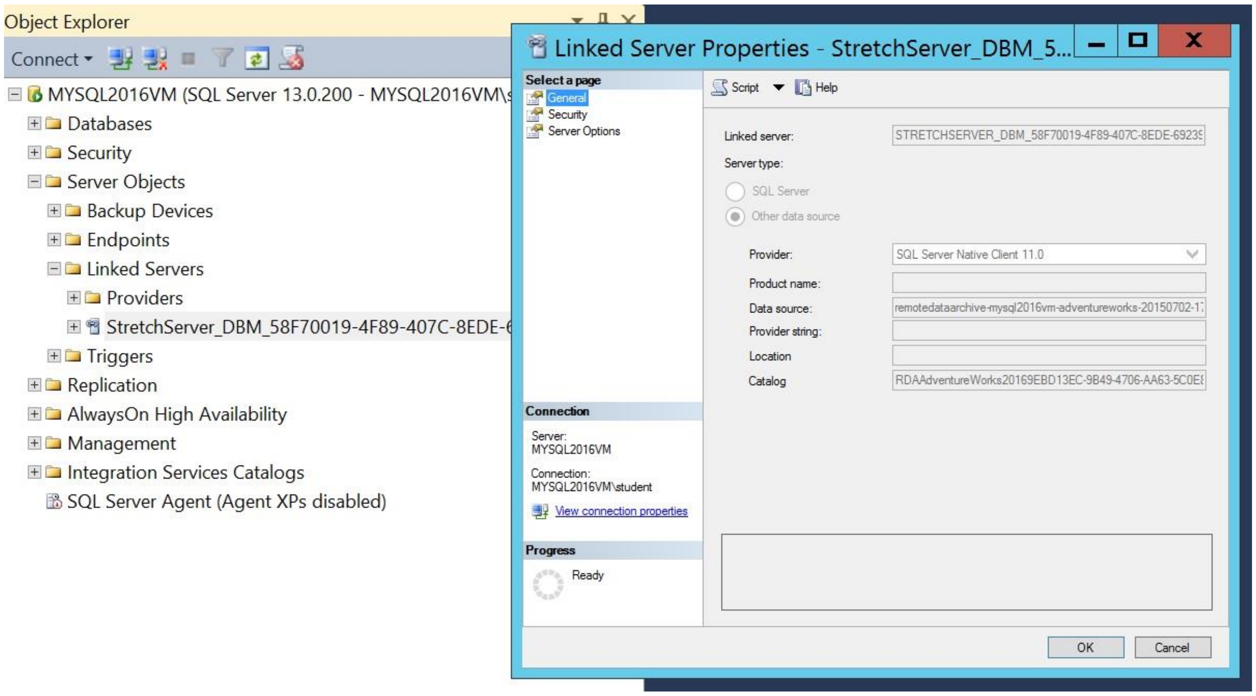
	object_id	remote_database_id	remote_table_name
1	1154103152	65536	Sales_SalesOrderDetail_1154103152_1FDE46DD-A82C-...

	table_id	database_id	migrated_rows	start_time_utc	end_time_utc	error_number	error_severity	error_state
1	1154103152	6	0	2015-07-02 17:46:27.240	2015-07-02 17:46:41.443	NULL	NULL	NULL
2	1154103152	6	10000	2015-07-02 17:46:41.473	2015-07-02 17:46:49.307	NULL	NULL	NULL
3	1154103152	6	10000	2015-07-02 17:46:49.307	2015-07-02 17:46:54.690	NULL	NULL	NULL
4	1154103152	6	10000	2015-07-02 17:46:54.690	2015-07-02 17:47:02.923	NULL	NULL	NULL
5	1154103152	6	10000	2015-07-02 17:47:02.923	2015-07-02 17:47:08.070	NULL	NULL	NULL
6	1154103152	6	10000	2015-07-02 17:47:08.070	2015-07-02 17:47:13.283	NULL	NULL	NULL
7	1154103152	6	10000	2015-07-02 17:47:13.283	2015-07-02 17:47:18.180	NULL	NULL	NULL
8	1154103152	6	10000	2015-07-02 17:47:18.180	2015-07-02 17:47:22.993	NULL	NULL	NULL

Query executed successfully. | MYSQL2016VM (13.0 CTP) | MYSQL2016VM\student (53)

Figure 14: System Catalog Views for Stretch Databases

The remote Azure SQL database, associated linked-server definition, remote endpoint, and appropriate Azure firewall settings are created automatically when the selected database is enabled as a Stretch Database. See Figures 15 and 16.


Object Explorer
Connect
MYSQL2016VM (SQL Server 13.0.200 - MYSQL2016VM\s
Databases
Security
Server Objects
Backup Devices
Endpoints
Linked Servers
Providers
StretchServer_DBM_58F70019-4F89-407C-8EDE-6
Triggers
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
Linked Server Properties - StretchServer_DBM_5...
Select a page
General
Security
Server Options
Script
Help
Connection
View connection properties
Progress
Ready
OK
Cancel

Linked server: STRETCHSERVER_DBM_58F70019-4F89-407C-8EDE-6923

Server type:

SQL Server [✓ Other data source]

Provider: SQL Server Native Client 11.0

Product name:

Data source: remotedataarchive-mysql2016vm-adventureworks-20150702-1

Provider string:

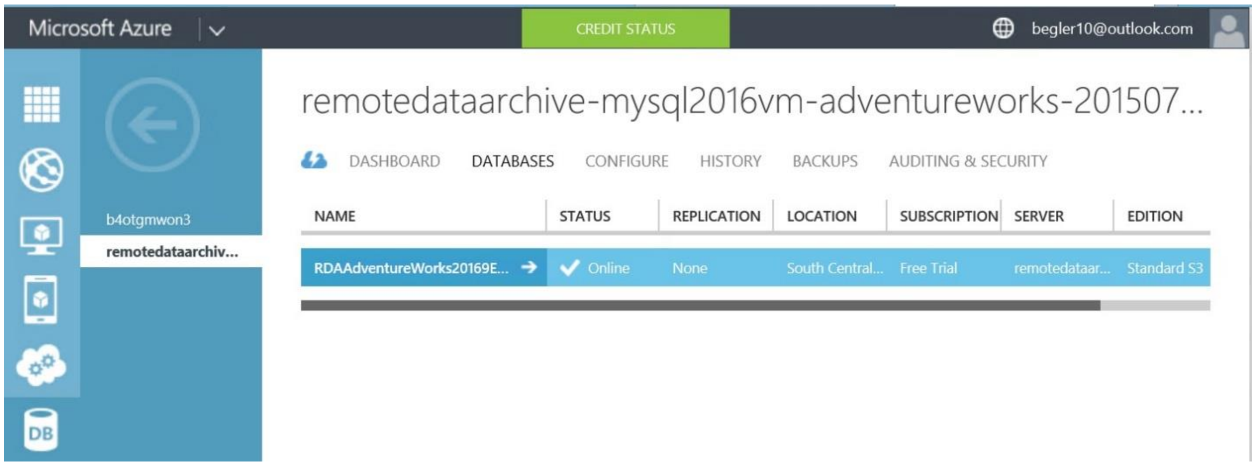
Location

Catalog RDAAdventureWorks20169EBD13EC-9B49-4706-AA63-5C0E

Server: MYSQL2016VM

Connection: MYSQL2016VM\student

Figure 15: Linked Server Definition for Remote Database



remotedataarchive-mysql2016vm-adventureworks-201507...

NAME	STATUS	REPLICATION	LOCATION	SUBSCRIPTION	SERVER	EDITION
RDAAdventureWorks20169E...	✓ Online	None	South Central...	Free Trial	remotedataar...	Standard S3

Figure 16: Azure SQL Database for Remote Database

With features such as this, Microsoft has successfully incorporated the cloud as an extension of the SQL Server architecture, rather than as a completely separate platform, and as such is allowing customers to leverage existing technologies in what is being termed the hybrid cloud.

Conclusion

Microsoft SQL Server 2016 has some great new features that will allow you to develop high-performing, more secure, scalable applications using the hybrid cloud. The fact that the features are largely incremental in nature should reassure users that Microsoft is building on the established foundation of SQL Server 2012 and 2014. Using similar architecture and management tools, customers will be able to smoothly upgrade their systems and skills based on the need for the new features, according to their own schedule.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

MCSA: SQL Server 2012 Boot Camp

MCSE: Data Platform Boot Camp

MCSE: Business Intelligence Boot Camp

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

Brian Egler is a Global Knowledge instructor and course director specializing in Microsoft SQL Server technologies. He currently resides in Highlands, North Carolina.