Intelligent Data Management with SQL Server

Session: 17

PolyBase, Query Store, and Stretch Database

Objectives

- Describe PolyBase
- Explain features and advantages of PolyBase
- Define and describe Query Store
- Explain how to dynamically stretch warm and cold transactional data from SQL Server to Azure
- Describe how to tune workload performance with Query Store

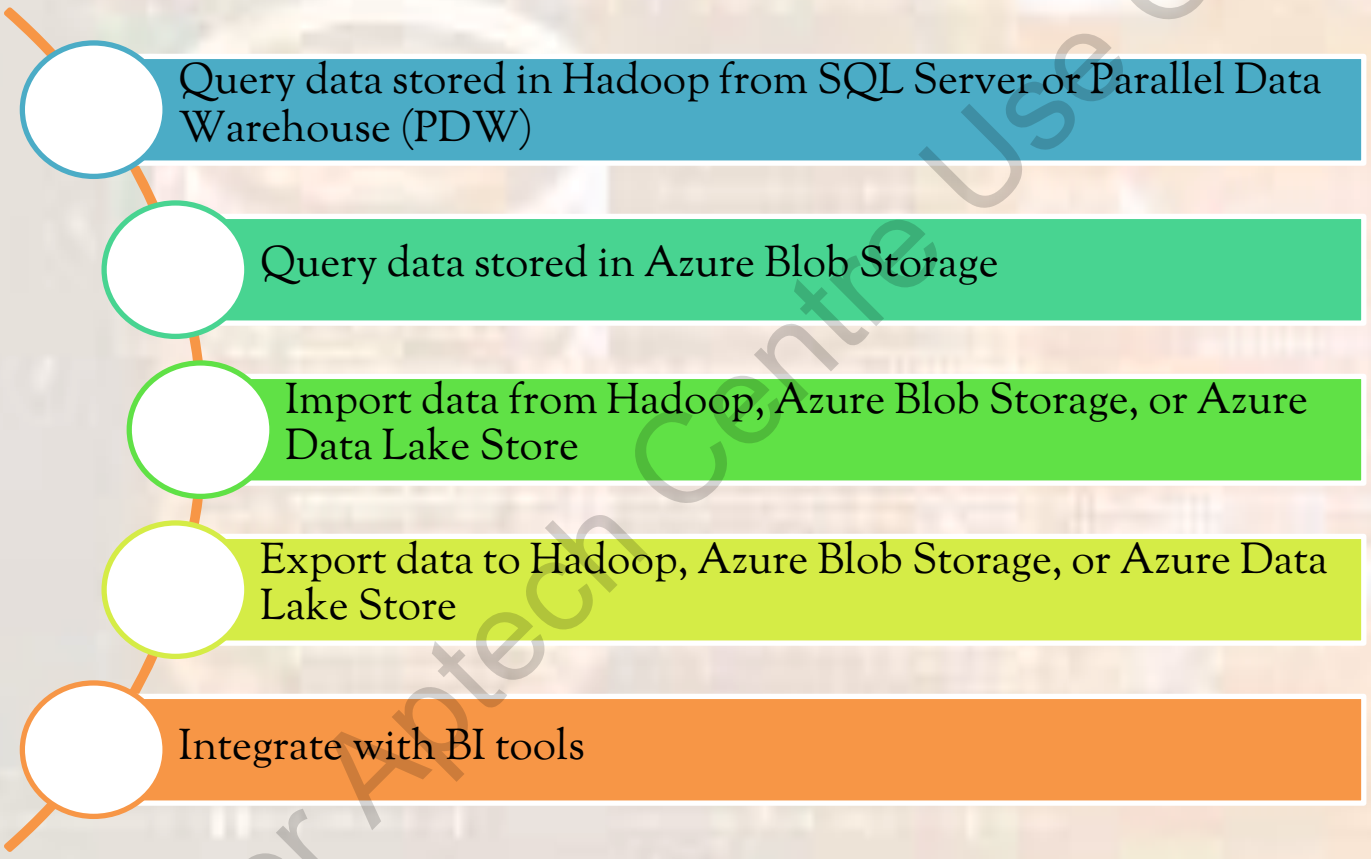
Understanding PolyBase 1-2

- PolyBase enables your SQL Server instance to process Transact-SQL queries that read data from external data sources.
- SQL Server 2016 and higher versions can access external data in Hadoop and Azure Blob Storage.

- Queries that access external data can also use to target relational tables in your SQL Server instance.
- This allows to combine data from external sources with high-value relational data in your database.

Understanding PolyBase 2-2

PolyBase enables the following scenarios in SQL server:



PolyBase Architecture 1-2

Architecture of PolyBase has some similarities to Hadoop architecture.

Head node

It contains SQL Server instance to which PolyBase queries are submitted. Each PolyBase group can have only one head node. A head node is a logical group of SQL Database Engine, PolyBase Engine, and PolyBase Data Movement Service on the SQL Server instance.

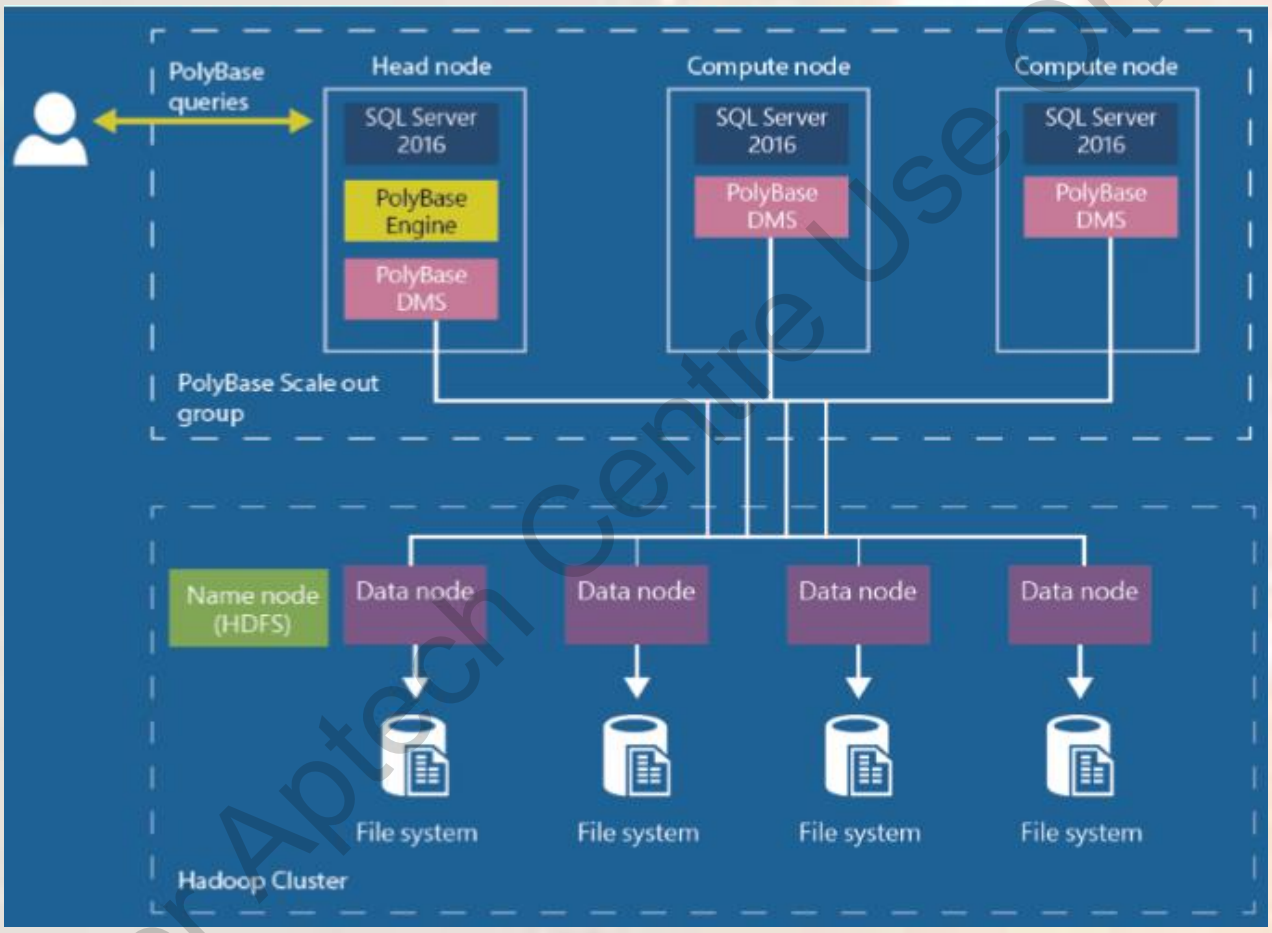
Compute node

It contains SQL Server instance that assists with scale-out query processing on external data. A compute node is a logical group of SQL Server and the PolyBase data movement service on SQL Server instance. A PolyBase group can have multiple compute nodes. Both head node and compute nodes must run on the same version of SQL Server.

Scale-out Reads

When querying external SQL Server, Oracle or Teradata instances, partitioned tables will benefit from scale-out reads. Each node in a PolyBase scale-out group can spin up to eight readers to read external data and each reader is assigned one partition to read in the external table.

PolyBase Architecture 2-2



PolyBase Architecture

Installing PolyBase in SQL Server 2019 1-4

Following are the pre-requisite to install PolyBase in SQL Server 2019:

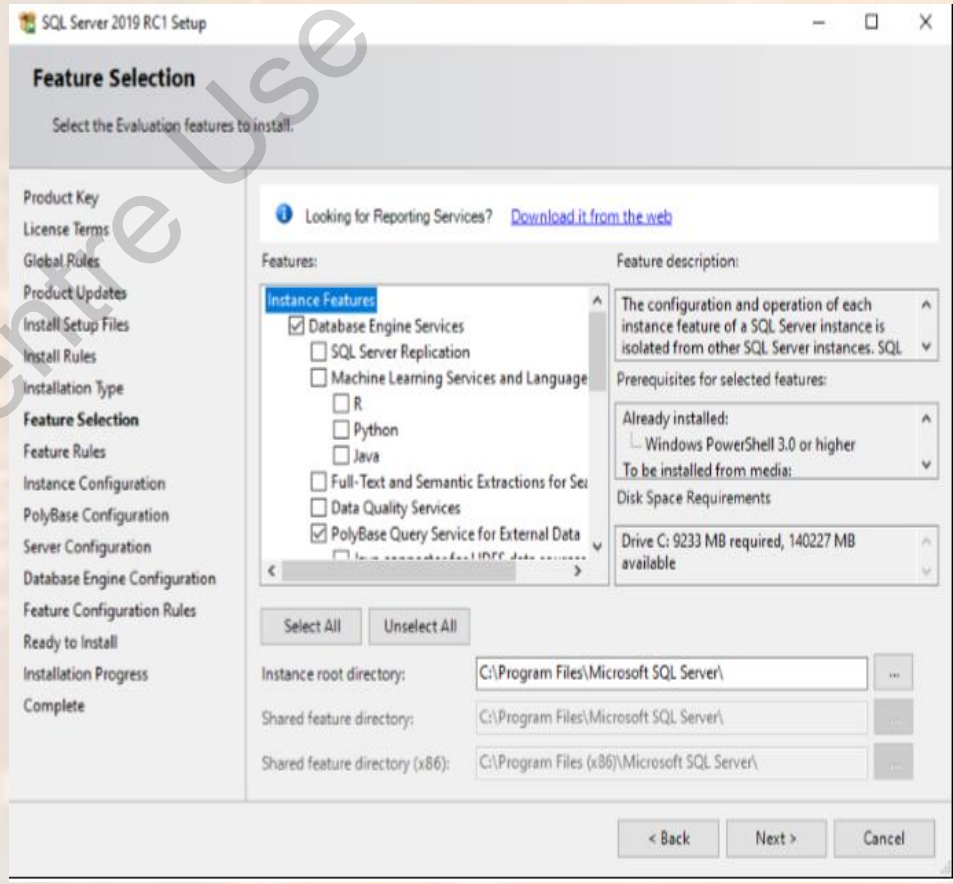
- 64-bit SQL Server Evaluation edition.
- Microsoft .NET Framework 4.5.
- Minimum memory: 4 GB.
- Minimum hard-disk space: 2 GB.
- Recommended: Minimum of 16 GB RAM.
- TCP/IP must be enabled for PolyBase to function correctly.

Installing PolyBase in SQL Server 2019 2-4

Single node or PolyBase scale-out group:

Following are the steps to install and set up PolyBase:

- Run the SQL Server setup.exe.
- Select Installation and then, select New standalone SQL Server installation or add features.
- On the Feature Selection page as shown in the figure, select PolyBase Query Service for External Data.



Installing PolyBase in SQL Server 2019 3-4

Enable PolyBase

- Once SQL Server installation is complete with the PolyBase feature, PolyBase must be configured to interact with external data sources.

Configure Hadoop Connectivity

- Run sp_configure with 'hadoop connectivity' and set an appropriate value for your provider.
- Restart SQL Server using services.msc.

SMS Agent Host	Provides ch...	Running	Automatic (D...	Local Syste
SNMP Trap	Receives tra...		Manual	Local Servic
Software Protection	Enables the ...		Automatic (D...	Network S..
Spot Verifier	Verifies pot...		Manual (Trig...	Local Syste
SQL Server (MSSQLSERVER)	Provides sto...	Running	Automatic	NT Service
SQL Server Agent (MSSQLSERVER)	Executes jo...		Manual	NT Service
SQL Server Browser	Provides SQ...		Disabled	Local Servic
SQL Server PolyBase Data Movement Service. (MSSQLSERVER)	Manages co...	Running	Automatic	Network S.
SQL Server PolyBase Engine (MSSQLSERVER)	Creates, co...	Running	Automatic	Network S.

SQL Server Services

Installing PolyBase in SQL Server 2019 4-4

Configure an external table

To query the data in your Hadoop data source, you must define an external table to use in Transact-SQL queries. Following steps describe how to configure the external table:

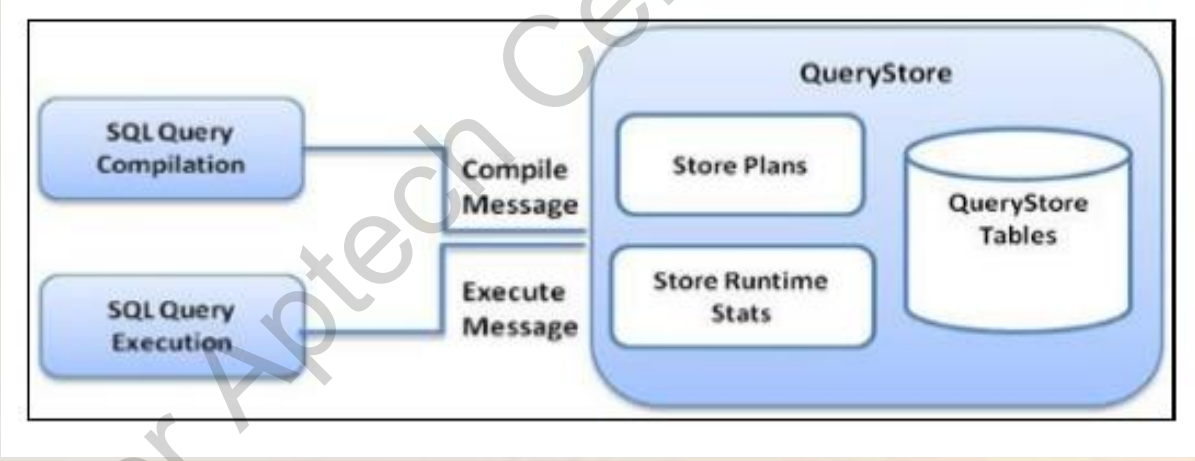
- Create a master key on the database
- Create a database scoped credential for Kerberos-secured Hadoop clusters
- Create an external data source with CREATE EXTERNAL DATA SOURCE
- Create an external file format with CREATE EXTERNAL FILE FORMAT
- Create an external table pointing to data stored in Hadoop with CREATE EXTERNAL TABLE

Query Store

- SQL Server Query Store feature provides an insight on query plan choice and performance.
- It simplifies performance troubleshooting by helping quickly to find performance differences caused by query plan changes.
- Query Store automatically captures a history of queries, plans, and runtime statistics, and retains these for review.
- It separates data as per time frames so database usage patterns can be identified and query plan changes happened on the server are noted.

Query Store Architecture

- Each query compilation or execution by SQL Server sends a message to the Query store.
- The information about the compilation and execution is first stored in cache and then, stored to the disk.



SQL Server Messages to Query Store

Enabling Query Store

- Query Store is not enabled by default for new SQL Server and Azure Synapse Analytics (SQL DW) databases and is enabled by default for new Azure SQL Database databases.

Using SQL Server Management Studio

1. In Object Explorer, right-click a database and then, click Properties.
2. In the Database Properties dialog box, select the Query Store page.
3. In the Operation Mode (Requested) box, select Read Write.

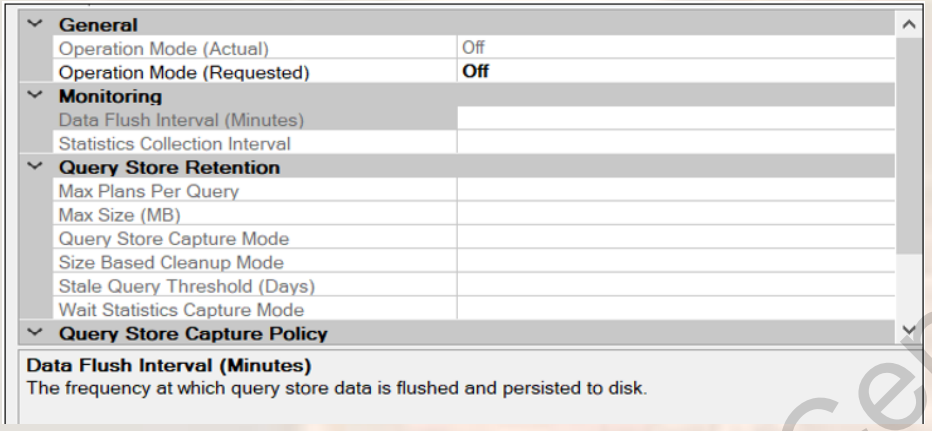
Using Transact-SQL

Use the ALTER DATABASE statement to enable the query store for a given database shown as follows:

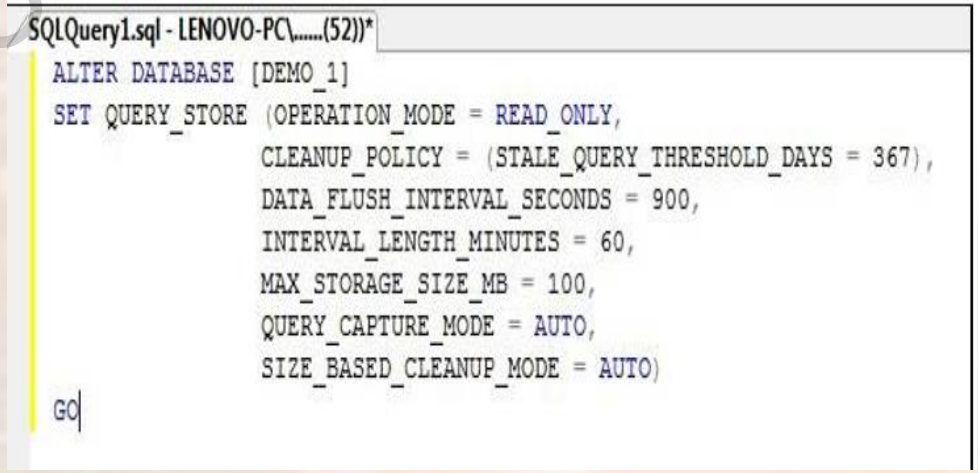
```
SET QUERY_STORE = ON (OPERATION_MODE = READ_WRITE);
```

Configuring Query Store

Once the Query Store feature is enabled, other parameters for Monitoring and Query Store Retention can be configured.



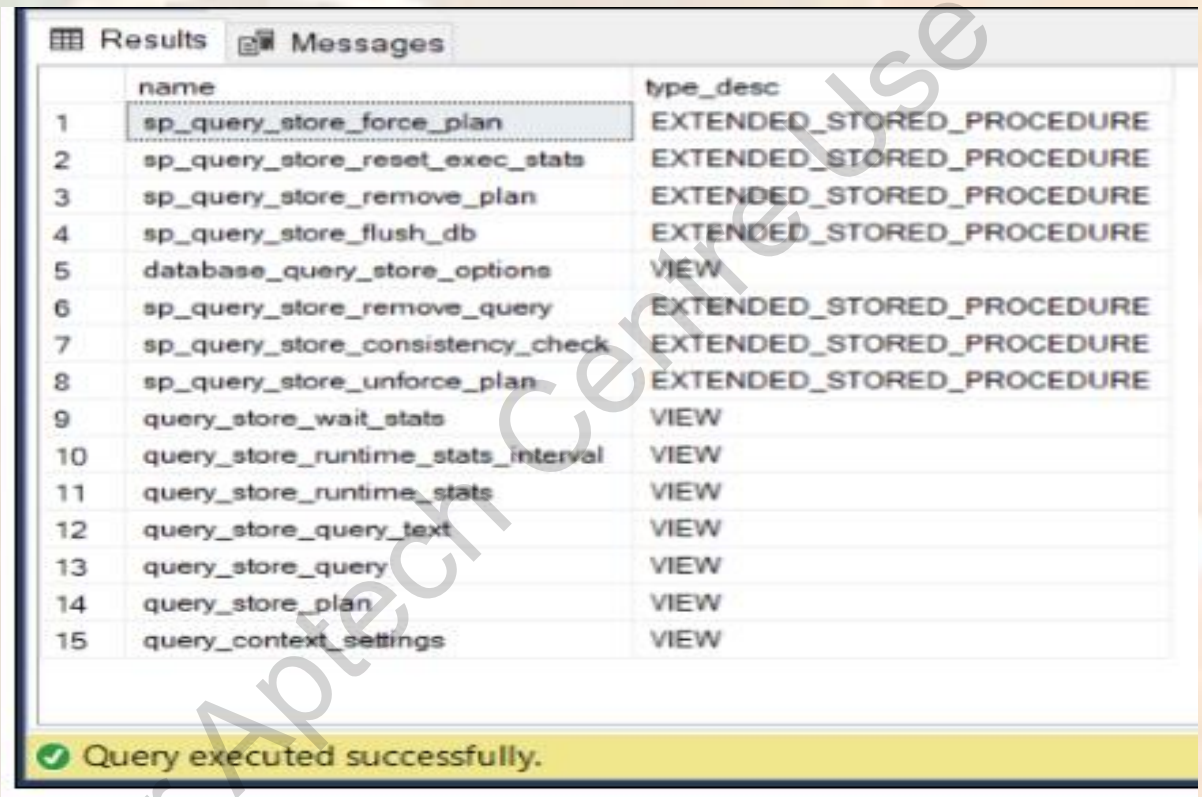
Configure Query Store Using SSMS



Configure Query Store Using Transact-SQL

Query Store System Objects

New system procedures and catalog views related to Query Store can be found in SQL Server.



System-Stored Procedures and Catalog Views

Performance Tuning with Query Store

If any recent plan changes have caused poor performance, Query Store can be used to quickly identify the issue and if required revert to a previous plan.

If any queries are consuming the most of system resources, such as CPU, memory, or IO they can be identified and accordingly changed to ensure optimal usage.

When changes to applications or platforms are planned, Query Store can be used to compare performance before and after change implementation. These changes may include installing new versions of applications, installing new hardware, compatibility level upgrades to the database, and adding or modifying indexes.

When upgrades are planned for Query Optimizer, performance of queries can be recorded before upgradation and fixed during upgradation.

When there is a need to optimize resources, queries with a lower frequency of execution can be identified and allocation of resources to them can be proactively handled.

Stretch Database

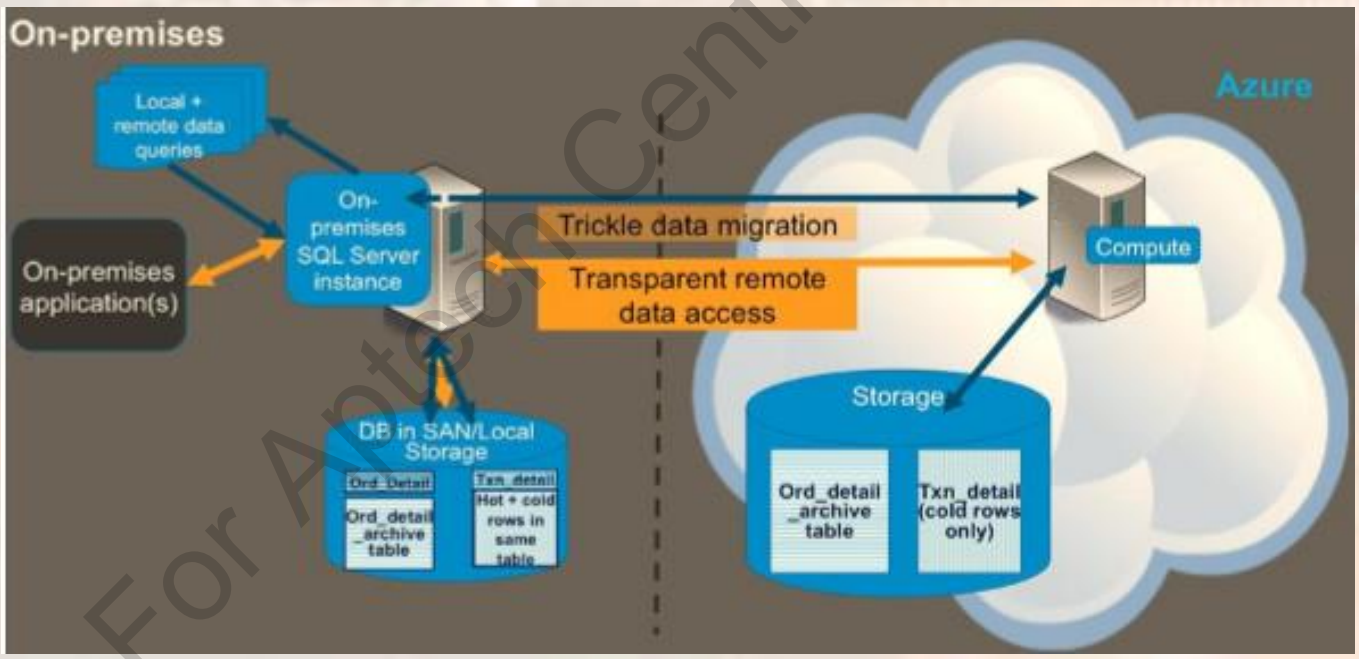
- Stretch Database is a new feature built into SQL Server that facilitates storage of a part of a database in the cloud.
- When there is a need to retain data for a longer time, Stretch Database feature enables secure migration of tables and data to the cloud.

Moreover, applications can still query the data in the same way as before. Enabling Stretch Database feature makes it possible to do the following:

- Move archived data as well as current data to the cloud securely by using the encryption features
- Access and query the data on the cloud at any time, without any changes to existing applications or queries
- Reduce storage requirements of on-premise data by using the vast storage capacity of the cloud
- Reduce processing burden on the on-premise data by running the processes on the cloud in a way that is transparent to the applications

Stretch Database Architecture

- A secure linked server definition is created in the on-premise SQL Server when the Stretch Database feature is enabled on a database.
- This establishes a connection to the remote or cloud database.
- When Stretch Database is enabled on a table, the resources on cloud are prepared and if migration is enabled, eligible data is migrated to the cloud database.

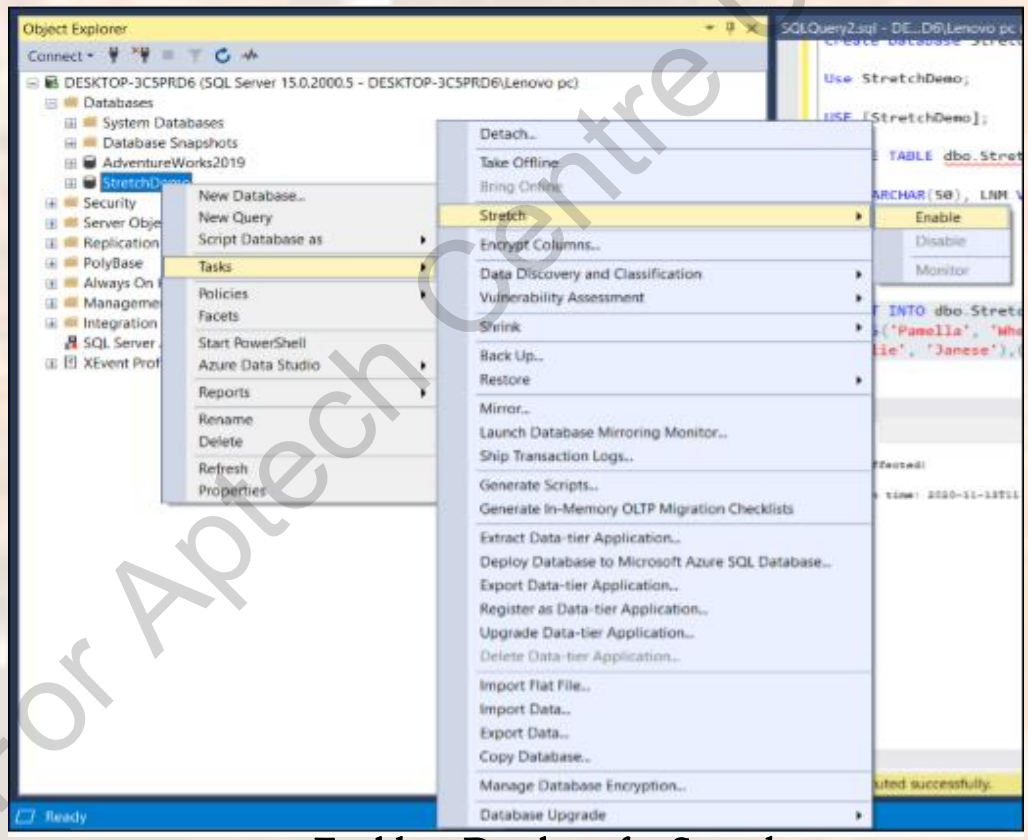


Stretch Database Architecture

Setting up Stretch Database 1-2

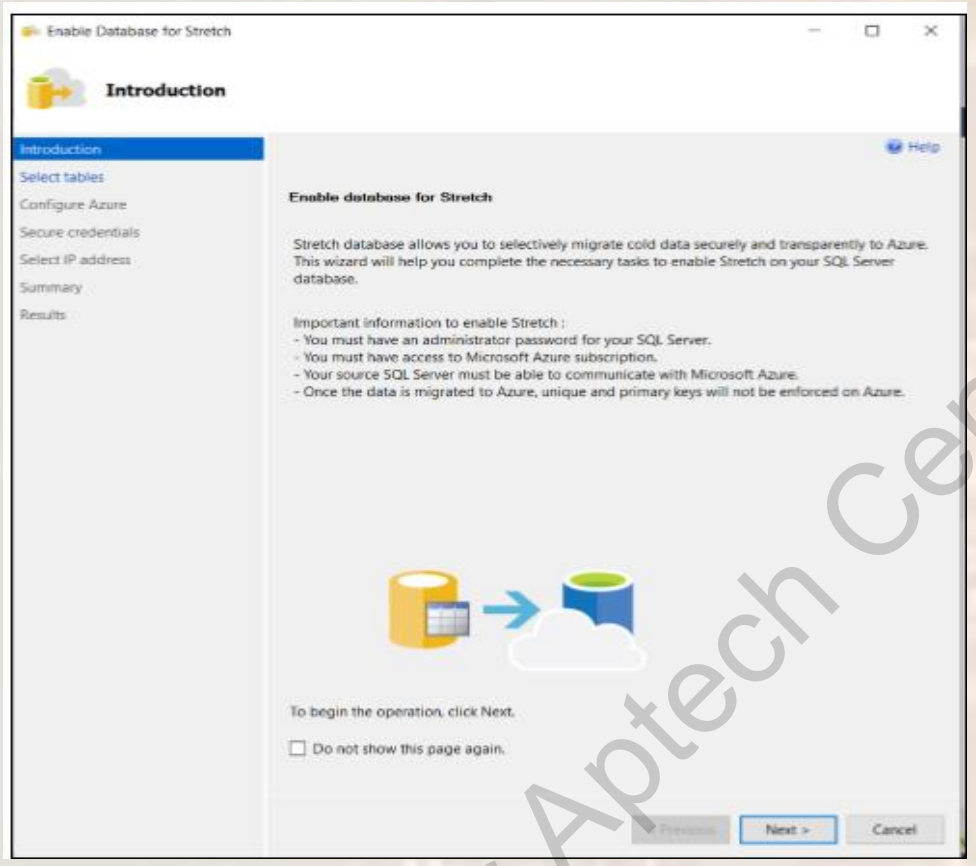
Prerequisite to Enable Stretch Database

- Enable Database for Stretch wizard configures the server for Stretch.
- An administrator has to enable it manually by running sp_configure first then, run the wizard, or an administrator has to run the wizard.

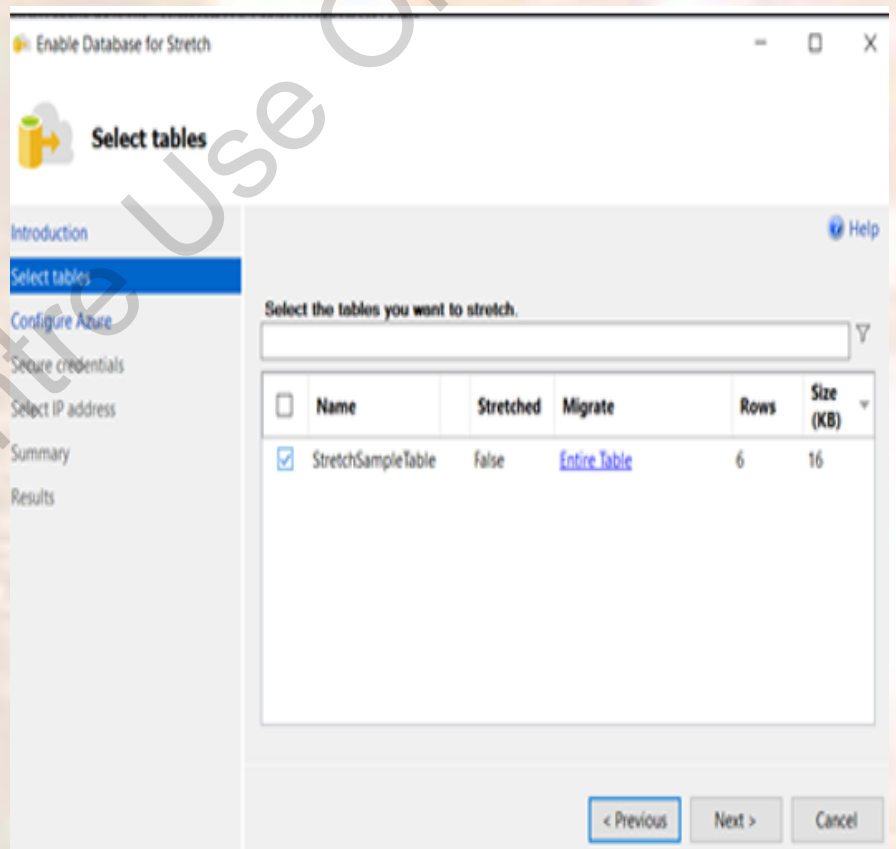


Enabling Database for Stretch

Setting up Stretch Database 2-2



Enable Database for Stretch



Select Tables

Stretch Database Limitations 1-3

Limitations for Stretch Enable table are:

Constraints

- Uniqueness is not enforced for UNIQUE constraints and PRIMARY KEY constraints in the Azure table that contains the migrated data.

DML operations

- You are not able to UPDATE or DELETE rows that have been migrated or rows that are eligible for migration, in a Stretch-enabled table or in a view that includes Stretch-enabled tables.
- You are not able to INSERT rows into a Stretch-enabled table on a linked server.

Indexes

- You are not able to create an index for a view that includes Stretch-enabled tables.
- Filters on SQL Server indexes are not propagated to the remote table.

Stretch Database Limitations 2-3

Following items currently prevent you from enabling Stretch for a table:

Table properties

- Tables that have more than 1,023 columns or more than 998 indexes
- FileTables or tables that contain FILESTREAM data
- Tables that are replicated, or that are actively using Change Tracking or Change Data Capture
- Memory-optimized tables

Data types

- text, ntext, and image
- timestamp
- sql_variant
- XML
- CLR data types including geometry, geography, hierarchyid, and CLR user-defined types

Column types

- COLUMN_SET
- Computed columns

Stretch Database Limitations 3-3

Constraints

- Default constraints and check constraints
- Foreign key constraints that reference the table. In a parent-child relationship (for example, Order and Order_Detail), you can enable Stretch for the child table (Order_Detail) but not for the parent table (Order).

Indexes

- Full text indexes
- XML indexes
- Spatial indexes
- Indexed views that reference the table

Summary

- The PolyBase feature provides seamless integration with external data sources, such as Hadoop or Azure Blob Storage.
- PolyBase eliminates the need to specialized skills on Hadoop internals by enabling query-runs on external data sources with simple Transact-SQL commands.
- Query Store is a built-in tool to improve performance by maintaining historical information of every query and execution plan.
- Query Store tracks the performance of queries and triggers alerts on poorly performing plans.
- Stretch Database in SQL Server 2019 enables stretching some part of a database to the Azure cloud, thereby, lowering long-term storage costs as well as maintenance efforts.
- Stretch Database is used to migrate archive and data in current transaction to the cloud securely.
- Stretch database is used to access cloud data at any time similar to the local data available on machines.