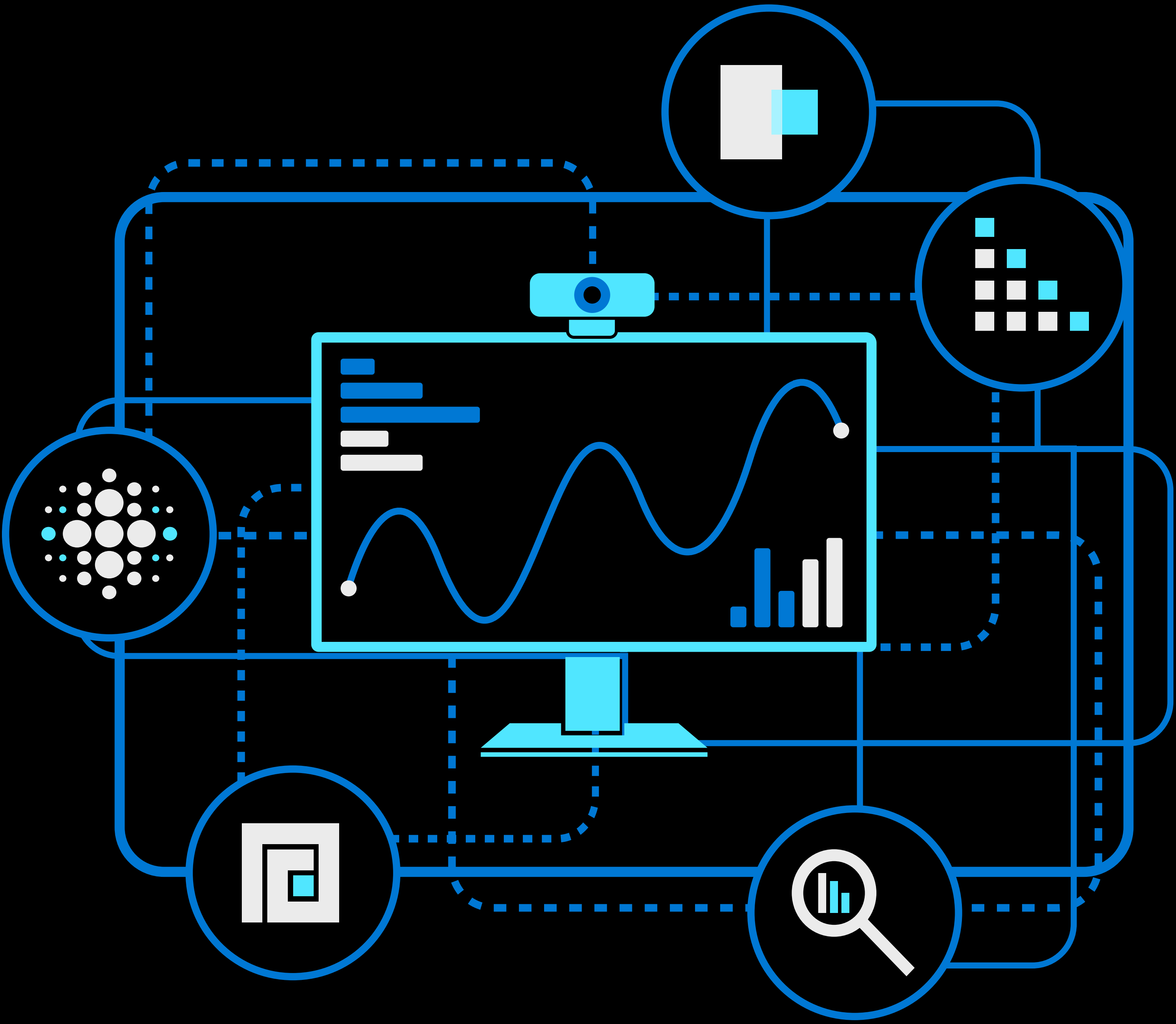
E-book series

Microsoft Azure

Five Steps to Simplify Your Data Mart and BI Solution



Five Steps to Simplify Your Data Mart and BI Solution

Overview

This book covers the basic steps to migrate your on-premises SQL data marts and OLAP workloads to Azure Synapse Analytics. You will learn how to incorporate data governance and data management using Azure Purview as a key part of your on-premises SQL modernization journey. Azure Purview enables you to scan, classify, and understand data in your on-premises SQL server to help you prepare and strategize for migration to the cloud. Then, you are going to learn the benefits of using Azure Synapse Analytics, which will cover data integration, data lake exploration, data warehousing, and cloud-scale machine learning for analytics. After migrating, you will learn how to easily search, interact, and analyze scanned data assets from within the Azure Synapse Studio.

In this book, we are going to cover the following topics that will help you modernize your data mart and BI solution:

- Before you start: govern and understand data in your on-premises SQL Server
- Getting started with your first Azure Synapse project
- Step 1: Serverless data lake exploration and streaming analytics
- Step 2: Date warehousing
- Step 3: Data integration
- Step 4: Optimizing data warehouse workloads for BI and reporting
- Step 5: Cloud-scale machine learning for analytics
- Integrate data analytics and data governance
- Additional resources

All the topics have been provided with reference links to dive deeper into any topic.
So, let’s start working toward modernizing your data marts.

Before you start: govern and understand data in your on-premises SQL Server

Before doing a migration project, it's important to first have a holistic understanding of the data you have captured and stored across your on-premises SQL Servers. This will help you plan and strategize how you will most effectively migrate different workloads and data streams. Data management and data governance should be a key first step in your migration journey and Azure Purview can help scan your SQL Server to provide a complete understanding of your data estate. Azure Purview is a unified data governance service that helps you manage and govern your on-premises, multicloud, and **software-as-a-service** (**SaaS**) data. You can easily create a holistic, up-to-date map of your data landscape with automated data discovery, sensitive data classification, and end-to-end data lineage.

Architecture and components

Azure Purview is designed to address the many data management and discovery challenges for data consumers, data producers, and security administrators. The product helps enterprises get the most value from their existing information assets.

Automatically scan metadata

The Azure Purview Data Map provides the foundation for data discovery and effective data governance. The Purview Data Map is a cloud native PaaS service that captures metadata about enterprise data present in analytics and operational systems both on-premises and in the cloud. The Purview Data Map is automatically kept up to date with a built-in automated scanning and classification system. Business users can configure and use the Purview Data Map through an intuitive UI and developers can programmatically interact with the Data Map using open-source Apache Atlas 2.0 APIs.

To learn more about how to automatically scan and classify data in your SQL Server using Azure Purview, refer to the following link: Register and scan an on-premises SQL server - Azure Purview | Microsoft Docs

Browse and search data

With the Azure Purview data catalog, business and technical users alike can quickly and easily discover relevant data using either search or browse. You can search with filters based on various lenses like glossary terms, classifications, sensitivity labels and more. To browse, you will click on the browse assets tile on the catalog home page, navigate to the source type tab, click on SQL Server, and navigate to the table that you're interested to learn more about. You can also browse for your SQL server tables by collection.

You can refer to the following links to learn more about how to search and browse data in the Azure Purview data catalog:

- How to search the Azure Purview data catalog
- How to browse the Azure Purview data catalog

Trace data lineage

Data consumers and producers can also visually trace the lineage of data assets starting from the operational systems on-premises through movement, transformation and enrichment with various data storage and processing systems in the cloud, to consumption in an analytics system like Power BI. When Azure Data Factory is used to move and transform your data, lineage relating to your SQL Server can automatically be captured and surfaced in Azure Purview.

To learn more about how you can link an Azure Data Factory instance to your Azure Purview account, refer to the following link: Connect to Azure Data Factory - Azure Purview | Microsoft Docs

Gain insights on data

With Azure Purview Data Insights, data officers and security officers can get a bird's eye view of all your SQL Server related insights around assets, scans, glossary, classification, and labels.

To learn more about how to use Azure Purview to gain insights on data, refer to the following link: Understand Insights reports in Azure Purview - Azure Purview | Microsoft Docs

Getting started with your first Azure Synapse project

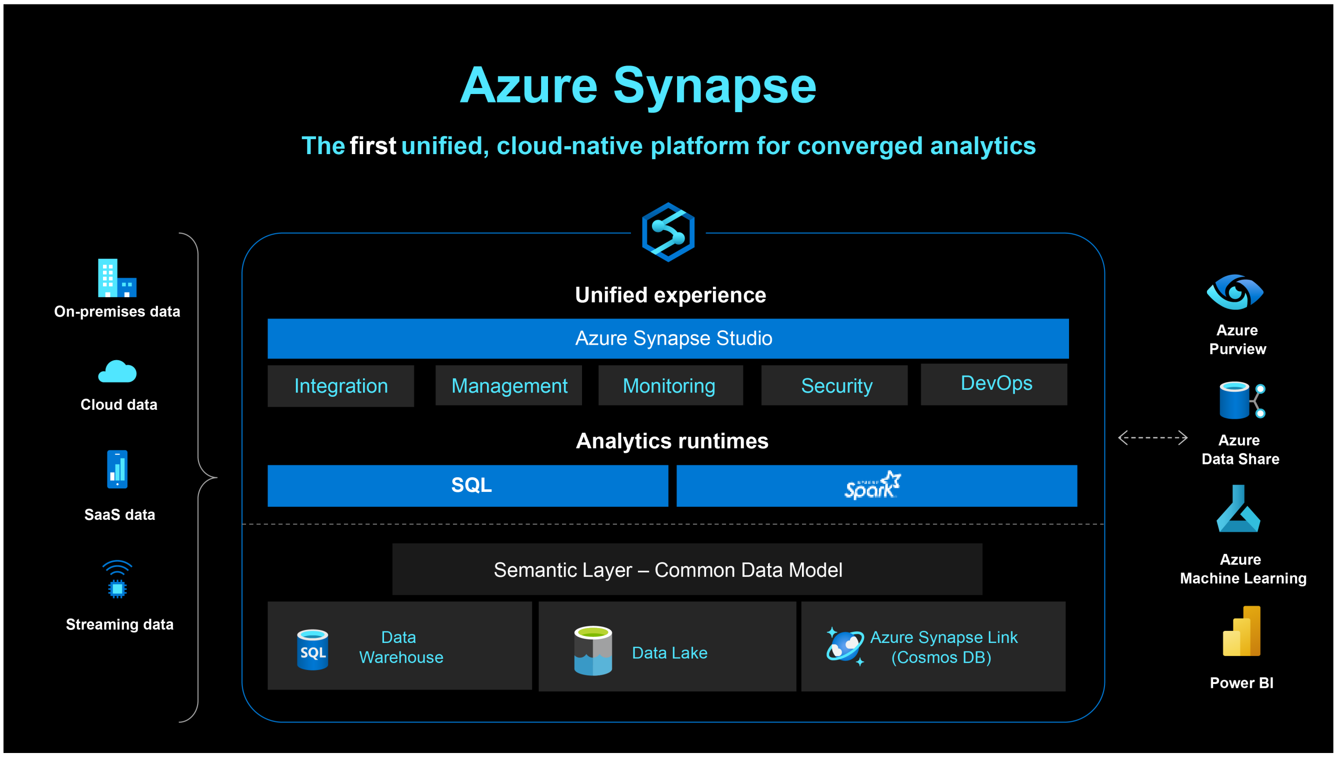
Azure Synapse is a limitless cloud service that allows you to stitch your enterprise data warehouse, data integration, and big data analytics together at scale. It allows you to query data using different compute engines and different languages.

Architecture and components

Azure Synapse is a unified analytics platform that brings together data integration, enterprise data warehousing, and big data analytics. It gives you the freedom to analyze data on your terms, using either serverless or dedicated options—at scale. Azure Synapse brings the divided worlds of data together with a unified experience to ingest, explore, prepare, transform, manage, and serve data for immediate BI and machine learning needs. The following is a list of all the components that we are going to cover in this book along with the associated workload type:

- SQL pool
- Spark pool
- Azure Storage
- Data ingestion capabilities for Azure Synapse
- Power BI
- Synapse Studio

In *Figure 1*, you can see how all the components are stitched together within Azure Synapse:



Azure Synapse

The first unified, cloud-native platform for converged analytics

On-premises data

Cloud data

SaaS data

Streaming data

Unified experience

Azure Synapse Studio

Integration | Management | Monitoring | Security | DevOps

Analytics runtimes

SQL | Spark

Semantic Layer – Common Data Model

Data Warehouse | Data Lake | Azure Synapse Link (Cosmos DB)

Azure Purview

Azure Data Share

Azure Machine Learning

Power BI

Figure 1: Components of Azure Synapse

Step 1: Serverless data lake exploration and streaming analytics

In this section, you are going to learn about the new serverless SQL pool for ad hoc analysis and data lake exploration.

The serverless SQL pool in Azure Synapse allows users to explore and query their data on the data lake using a familiar T-SQL syntax without moving the data to a relational data store. You can perform simple and scalable data transformations in the data lake to generate meaningful insights. This data can be loaded to a dedicated SQL pool for data warehousing workloads and can also be fed to Power BI.

If you come from a SQL Server background, you will know that data needs to be copied from the data lake to relational databases if you need to derive any insight or perform any transformation on your data. However, in the case of a serverless SQL pool, you may not need to relocate your data to explore it. Following are a few advantages that could be interesting use cases for a serverless SQL pool:

- Quickly explore the data in various formats (Parquet, CSV, JSON) in its data lake directly, to derive insights from it.
- Create a logical data warehouse without relocating data anywhere from your data lake.
- Generate Power BI reports directly using your data in a data lake.
- Easily transform your data using T-SQL and ingest it to a relational data store.

To learn more about the benefits of serverless SQL pools, refer to the following link: Serverless SQL pool in Azure Synapse Analytics.

Step 2: Data warehousing

A SQL pool is used to store relational data to run analytics at a massive scale in the cloud. It uses a columnar storage format to reduce data storage costs and improve query performance. In a dedicated SQL pool, compute is kept separate from storage so that you can scale your compute engine as per requirements, independent of your data in the system.

A SQL pool leverages a **Massive Parallel Processing** (**MPP**) engine to distribute the computational processing of your data across multiple compute nodes. All these compute nodes receive their instructions from a control node, as can be seen in *Figure 2*:

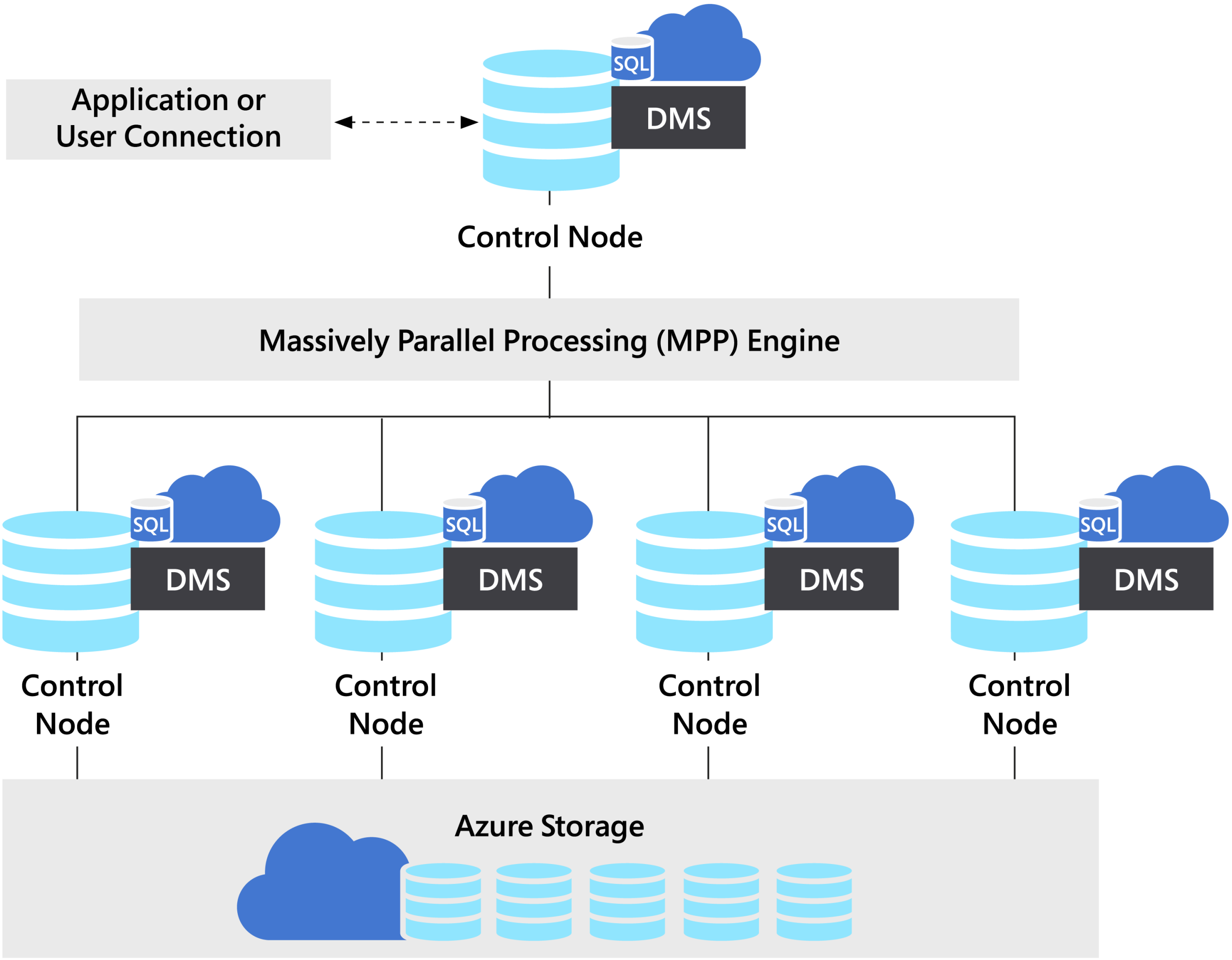


Figure 2: Architecture for a dedicated SQL pool in Azure Synapse

User data is stored in Azure Storage, which can be accessed by the compute nodes. An internal service, also known as a **Data Movement Service** (**DMS**), is used to move data across nodes to run the user queries in parallel.

To learn more about this architecture, refer to the following link: Dedicated SQL pool (formerly SQL DW) architecture in Azure Synapse Analytics.

You can go through three different stages for migrating your on-premises SQL data marts to a SQL pool:

- Metadata migration using Azure Synapse Pathway
- Table design and code changes
- Migrating data to Azure Synapse using data integration capabilities

In this section, we will learn about metadata migration and code changes. We will cover data migration in the *Step 3: Data integration* section later on.

Technical requirements

The following are the technical requirements that you need to perform before migrating your SQL data warehouse to a dedicated SQL pool:

1. Create your Synapse workspace.
2. Create a SQL pool in your Synapse workspace.

After creating the Synapse workspace and SQL pool, the first step is metadata migration, which can be performed using the Azure Synapse Pathway Microsoft tool. Next, we need to make some code changes apt for the MPP engine, and, last but not least, we will migrate our data using Synapse pipelines. So, let's go through all these steps to start our migration journey.

Metadata migration using Azure Synapse Pathway

For most organizations, migrating their data marts to a new analytics solution can be complicated. Integrating siloed data across business units for migration comes with many challenges because business logic and other database objects need to be disentangled when that data moves. One of the most complex migration tasks is translating SQL code. Converting hundreds of lines of legacy code across database objects requires data teams to manually rewrite existing code or hire **system integrators** (**SIs**) to complete the task.

Organizations have sprints requiring thousands of hours of intensive labor and knowledge transfer, which are prone to human error and incur large costs. However, Azure Synapse introduced Azure Synapse Pathway, which can be used to translate all SQL code automatically. The traditional process of manual code conversion can now be automated in a fraction of the time; all while cutting out errors and reducing the total cost of the migration.

Most of the scripts can be run on Synapse SQL without any modification. However, certain keywords are not supported in Synapse SQL. It could be a difficult task to identify all the scripts that need to be modified and start making the required changes one after the other. However, you can download Azure Synapse Pathway to migrate all your objects directly to Synapse SQL.

You may get some error messages for some of the scripts where manual intervention might be required, but this tool reduces the manual effort to a great extent.

Go through the following steps to migrate all the schemas and objects from an on-premises data warehouse to Azure Synapse:

1. Download Azure Synapse Pathway.
2. Follow the instructions to install this tool on your machine.
3. Before we can start using this tool, we need to generate scripts for all the database objects from the on-premises data warehouse. Connect to your data warehouse using the SQL Server Management Studio tool.

4. Right-click on your data warehouse and click on **Tasks** > **Generate Scripts…**:

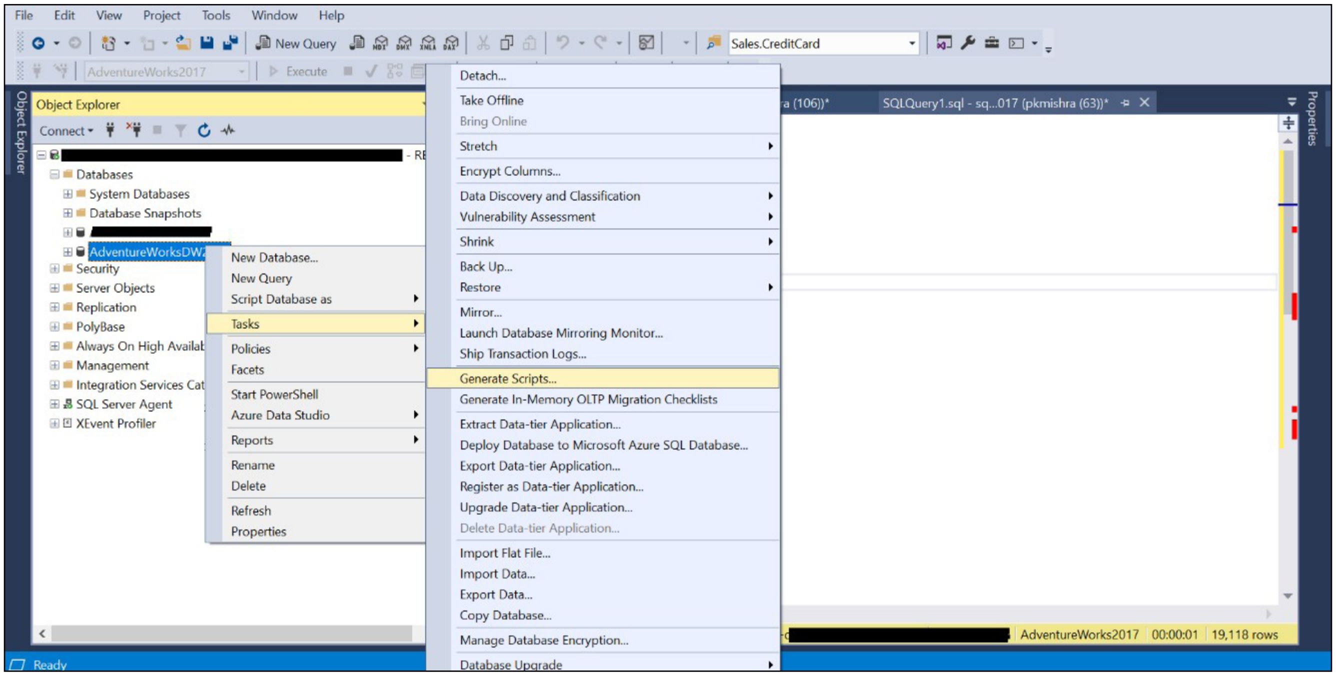
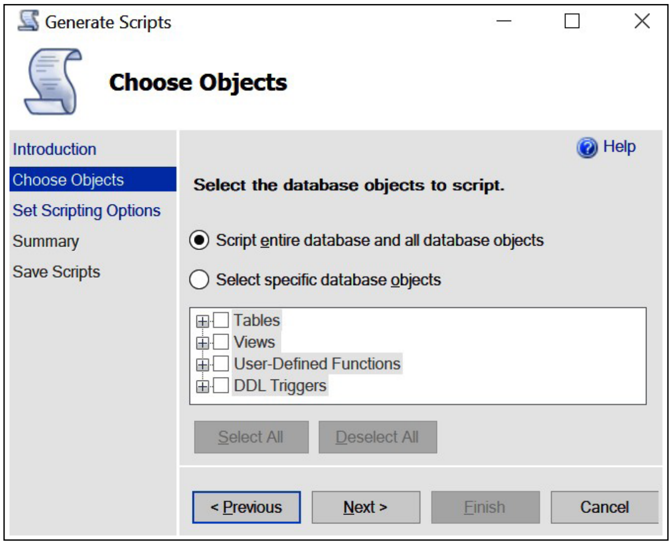


Figure 3: Generating scripts for the database objects

5. Go through the details provided on the **Introduction** page and then click on **Next**.
6. On the next screen, you can select the radio button to **Script entire database and all database objects** or you can click on another radio button to **Select specific database objects**. In this example, let's select the first radio button to select all the database objects:



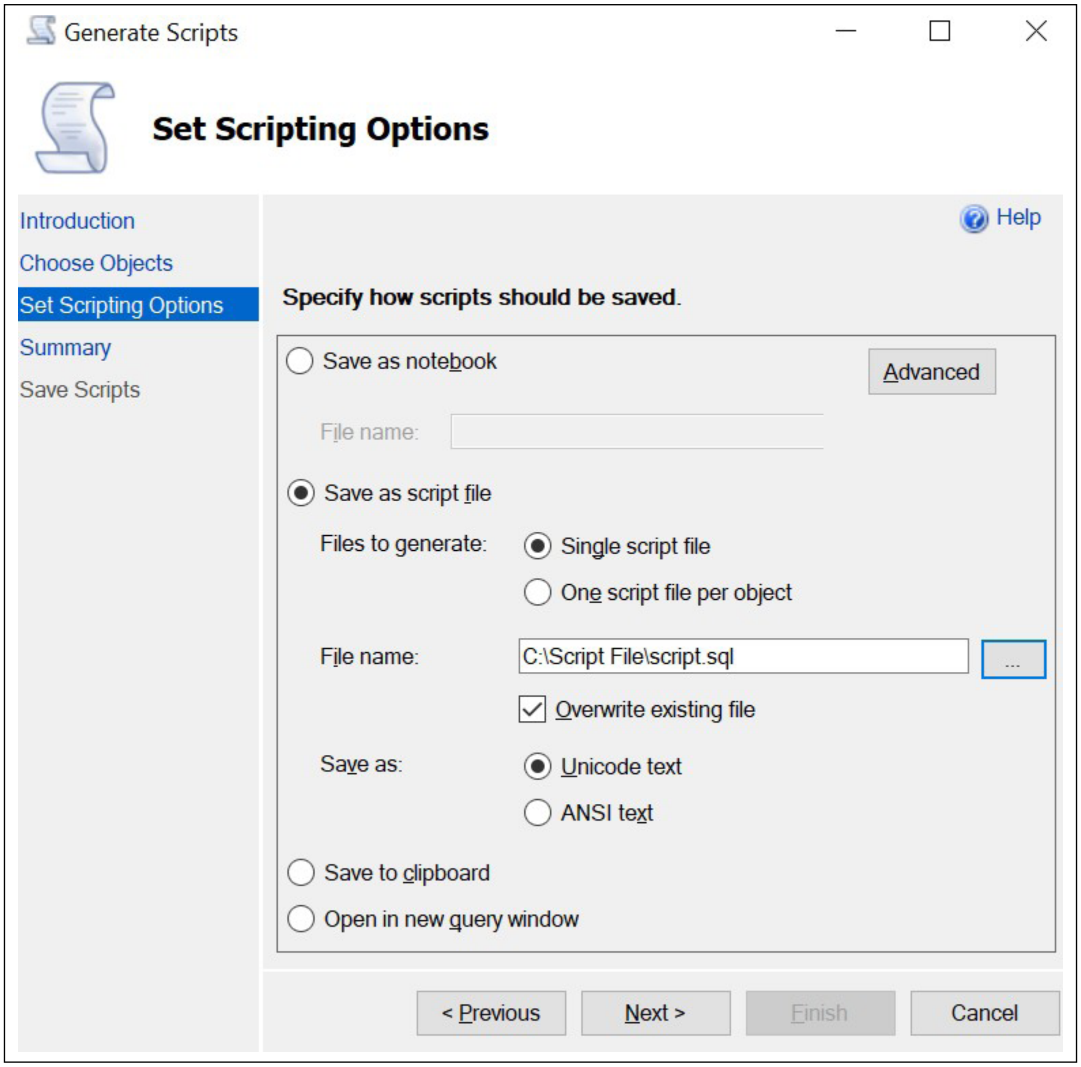
Select the database objects to script.

● Script entire database and all database objects

○ Select specific database objects

Figure 4: Selecting database objects to generate scripts

7. On the next screen, select the radio button **Save as script file** and provide an appropriate file path to save the generated scripts:



Generate Scripts

Set Scripting Options

Specify how scripts should be saved.

○ Save as notebook

File name:

● Save as script file

Files to generate: ● Single script file

○ One script file per object

File name: C:\Script File\script.sql

☑ Overwrite existing file

Save as: ● Unicode text

○ ANSI text

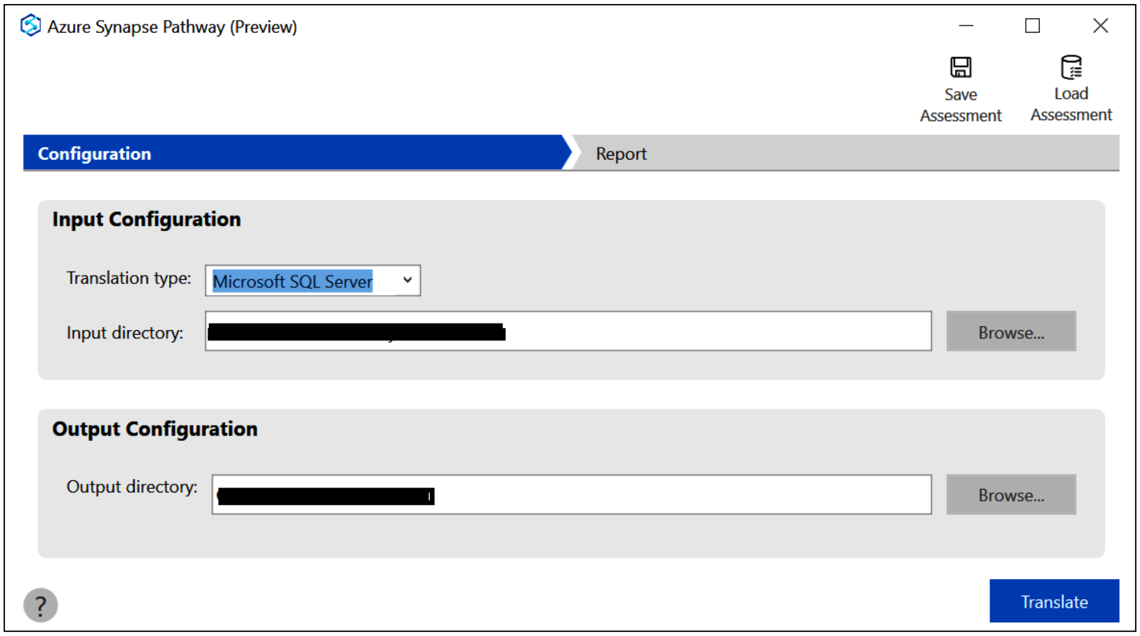
○ Save to clipboard

○ Open in new query window

Figure 5: Providing the file name to save the generated SQL script

8. Click on **Next** to review your selections and again click on **Next** after reviewing all the details.
9. In a couple of minutes, your new script file will be generated and saved in the required location. Click on **Finish** to close the **Generate Scripts** window.
10. Now, open the **Azure Synapse Pathway** tool on your machine and select **Microsoft SQL Server** from the drop-down list of **Translation type**.
11. Click on the **Browse…** button to select the input directory where you saved the database script file generated in *step 9*.

12. Then, click on the **Browse...** button to select the output directory where you want to store your converted script and click on **Translate**:



Input Configuration

Translation type: Microsoft SQL Server

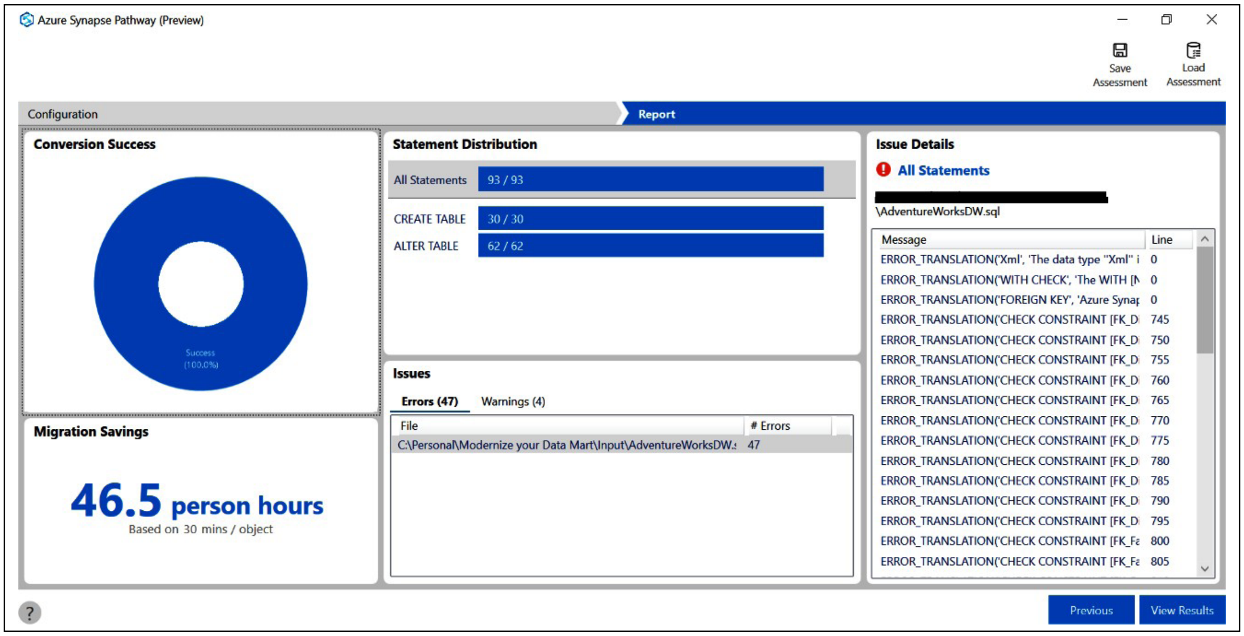
Input directory: [redacted]

Output Configuration

Output directory: [redacted]

Figure 6: Providing the configuration for Azure Synapse Pathway

13. It may take a couple of minutes to complete the migration process. You can see the migration report on the **Azure Synapse Pathway** tool once the migration is complete:



Conversion Success

Migration Savings

46.5 person hours
Based on 30 mins / object

Statement Distribution

All Statements	93 / 93
CREATE TABLE	30 / 30
ALTER TABLE	62 / 62

Issues

File	# Errors
C:\Personal\Modernize your Data Mart\Input\AdventureWorksDW.:	47

Issue Details

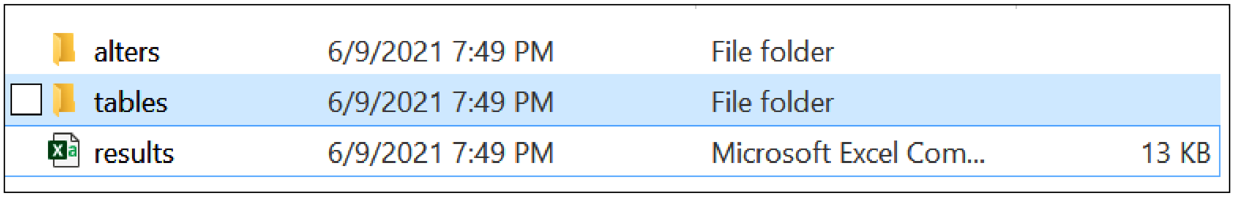
All Statements

[redacted]
\AdventureWorksDW.sql

Message	Line
ERROR_TRANSLATION('Xml', 'The data type "Xml" i	0
ERROR_TRANSLATION('WITH CHECK', 'The WITH [N	0
ERROR_TRANSLATION('FOREIGN KEY', 'Azure Synap	0
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	745
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	750
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	755
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	760
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	765
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	770
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	775
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	780
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	785
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	790
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Di	795
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Fa	800
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_Fa	805

Figure 7: A snapshot of the migration report generated by Azure Synapse Pathway

14. Click on **View Results** to go to the output directory specified in the previous step, and you'll see the translated script file(s) based on your input directory structure:



	alters	6/9/2021 7:49 PM	File folder	
□	tables	6/9/2021 7:49 PM	File folder	
	results	6/9/2021 7:49 PM	Microsoft Excel Com...	13 KB

Figure 8: A snapshot of the tables folder and results Excel sheet generated in the output directory

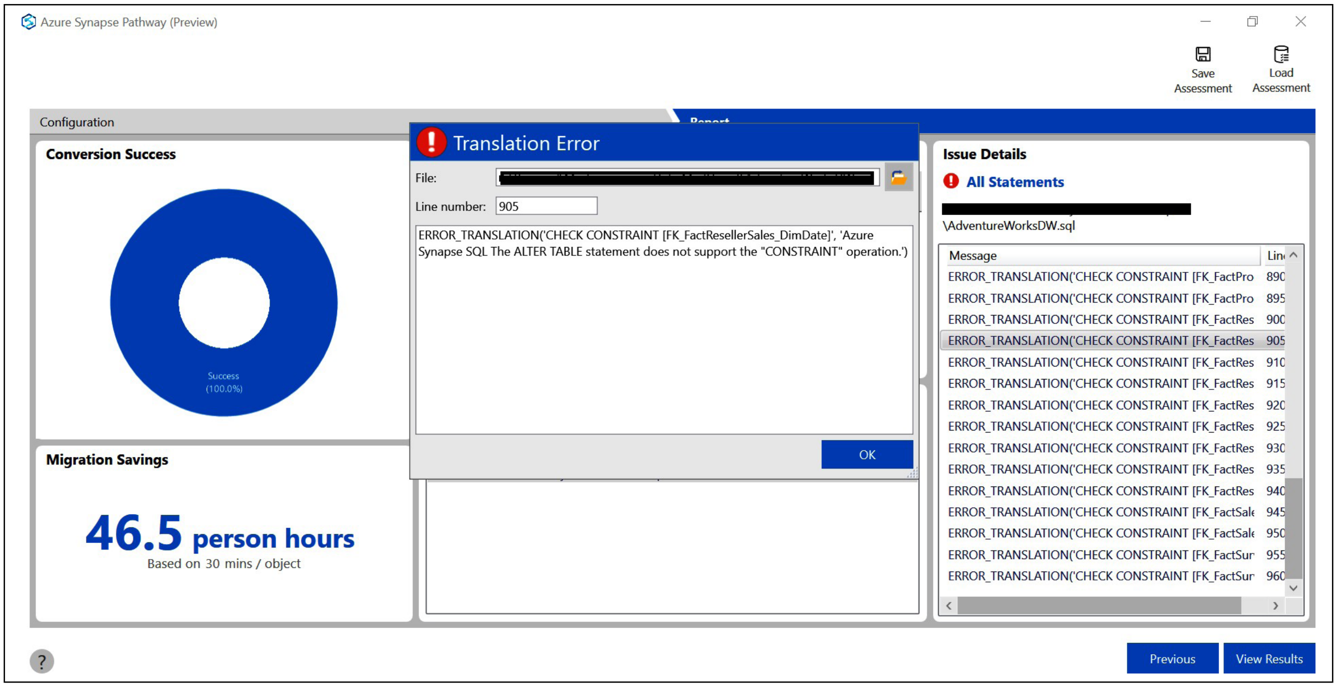
Now that we have successfully migrated all the database objects, let's see whether we need to make some table design or code changes.

Table design and code changes

Since we learned earlier in this chapter that SQL pools leverage MPP-based engines, it would be a wise decision to redesign your tables to have a correct distribution type for your table. Correct distribution will help you to achieve optimized performance on SQL pools. You can get guidance for designing distributed tables using a dedicated SQL pool by referring to the documentation.

After making all the necessary changes to the table designs, we need to address the error messages thrown by the Azure Synapse Pathway tool. Some code might not be compatible with Synapse SQL pools, so we need to make the required changes to make the code compatible with SQL pools.

Figure 9 shows one of the error messages that need to be addressed before executing it on a SQL pool:



Translation Error

File: [redacted]

Line number: 905

ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactResellerSales_DimDate]', 'Azure Synapse SQL The ALTER TABLE statement does not support the "CONSTRAINT" operation.')

Issue Details

All Statements

[redacted]
\AdventureWorksDW.sql

Message	Lin
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactPro	890
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactPro	895
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	900
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	905
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	910
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	915
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	920
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	925
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	930
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	935
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactRes	940
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactSale	945
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactSale	950
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactSur	955
ERROR_TRANSLATION('CHECK CONSTRAINT [FK_FactSur	960

Figure 9: A snapshot of one of the error messages on Azure Synapse Pathway

Although a SQL pool supports most of the aspects of T-SQL stored procedures, certain aspects are not implemented in a dedicated SQL pool. You can find the list here.

There are some development recommendations and coding techniques that you must consider while migrating the objects from on-premises SQL data marts.

Like a stored procedure, certain recommendations need to be considered before creating tables in Azure Synapse. A SQL pool stores and operates on tables across distributions. In Azure Synapse, you have three methods for distributing the data: round-robin (default), hash, and replicated. Users can decide which distribution method will be apt for any specific table as per the guidelines provided here.

Step 3: Data integration

Azure Synapse has data integration capabilities to modernize your ETL/ELT processes. We need to create linked services to set up the connection with the source and the target before adding any activity to the pipeline. Users can add different types of activities to the pipeline for data engineering and later these pipelines can be scheduled to run at regular intervals.

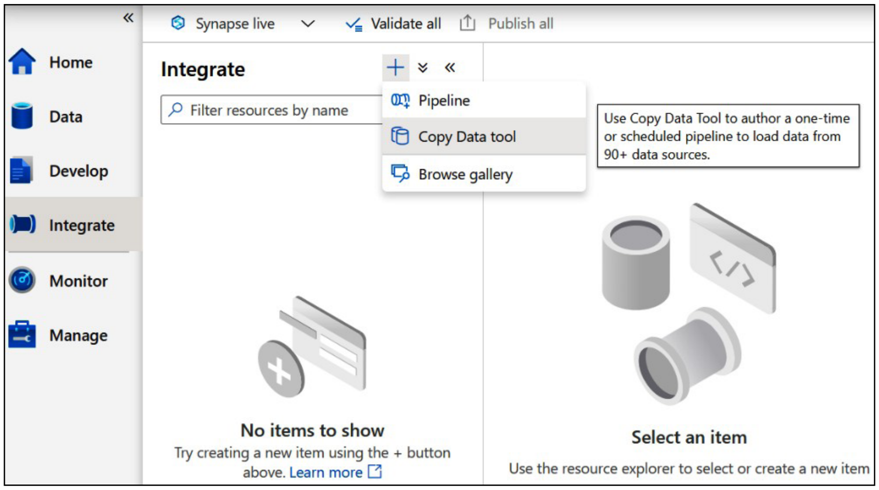
We are going to learn in this chapter about using the COPY activity with pipelines to bring data to Azure Synapse. However, you can explore Azure Synapse pipelines in much more detail by going through the Integrate with pipelines documentation.

Azure Synapse has made it easy for users to bring their data integration workloads from on-premises within the Synapse Studio to Azure Synapse by integrating **pipelines** within **Synapse Studio**. Your ETL/ELT jobs can be done using a drag-and-drop, low-code interface as soon as your Azure Synapse workspace is provisioned.

Important: SSIS Integration Runtime is not currently supported in Azure Synapse. However, you can create data integration pipelines within Azure Synapse to replicate the functionalities of your SSIS package. This feature is available in Azure Data Factory and you can learn more about it by going through Create an Azure-SSIS integration runtime in Azure Data Factory.

You can go to the **Integrate** hub in Synapse Studio to create or edit a pipeline as seen in *Figure 10*. You can also browse through the gallery to start playing with pipeline templates. Now, let's go through the following steps to copy the data from an on-premises SQL data warehouse to the dedicated SQL pool on Azure:

1. Go to the **Integrate** hub of your Synapse Studio, click on **+**, and select **Copy Data tool**, as seen in *Figure 10*:



Use Copy Data Tool to author a one-time or scheduled pipeline to load data from 90+ data sources.

No items to show

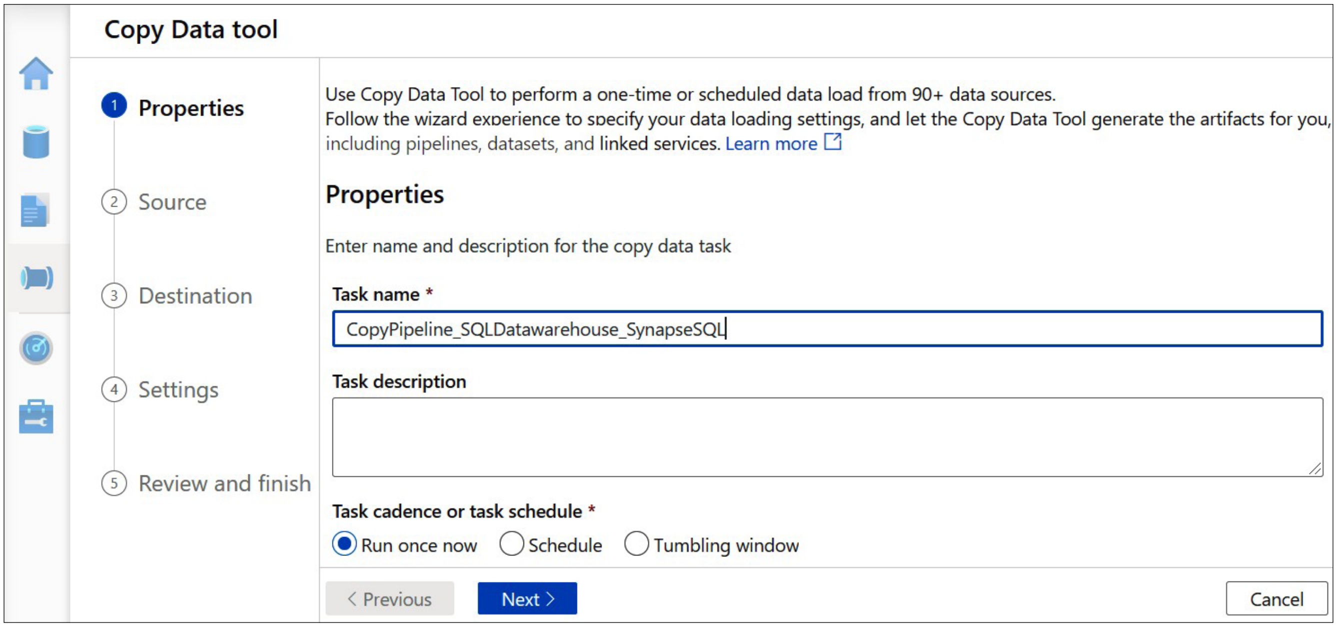
Try creating a new item using the + button above. Learn more

Select an item

Use the resource explorer to select or create a new item

Figure 10: A snapshot of the Integrate hub of Synapse Studio

2. Provide an appropriate name in the corresponding field, leave the default value for **Task cadence or task schedule**, and then click on **Next**:



Copy Data tool

Use Copy Data Tool to perform a one-time or scheduled data load from 90+ data sources.
Follow the wizard experience to specify your data loading settings, and let the Copy Data Tool generate the artifacts for you, including pipelines, datasets, and linked services. Learn more

Properties

Enter name and description for the copy data task

Task name *
CopyPipeline_SQLDatawarehouse_SynapseSQL

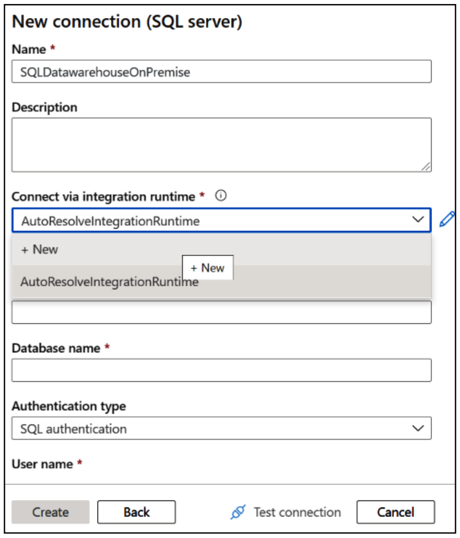
Task description

Task cadence or task schedule *
● Run once now ○ Schedule ○ Tumbling window

Figure 11: Setting up the properties for the Copy Data tool on Synapse Studio

3. Next, we need to set up a connection for the on-premises SQL data warehouse. So, provide an appropriate name for this connection.

4. Then, we need to create a self-hosted integration runtime to establish a connection between the on-premises environment and Azure. So, click on **+ New** appearing in the drop-down list for **Connect via integration runtime**:



New connection (SQL server)

Name *

SQLDatawarehouseOnPremise

Description

Connect via integration runtime * ⓘ

AutoResolveIntegrationRuntime

Database name *

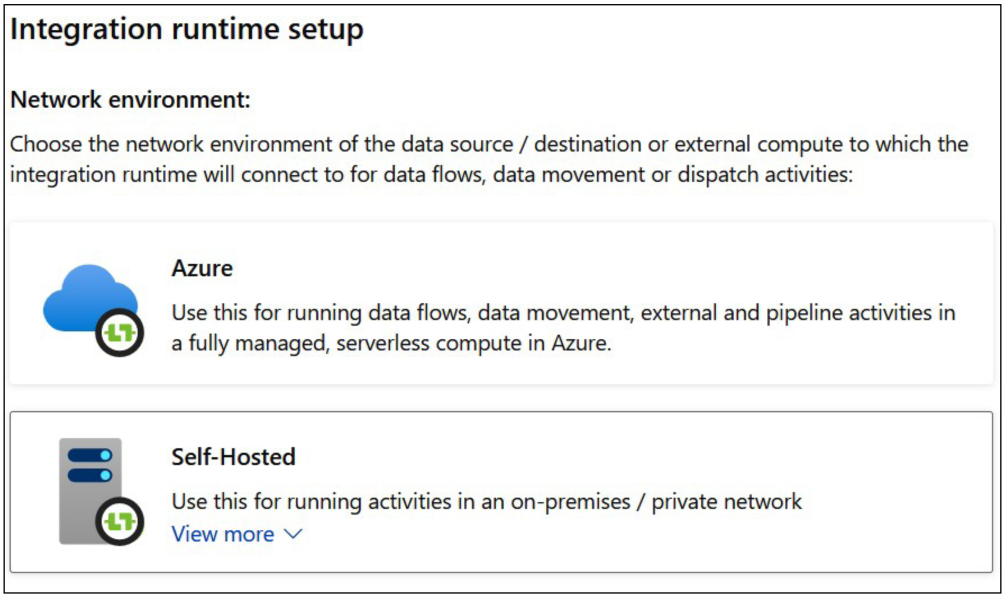
Authentication type

SQL authentication

User name *

Figure 12: Creating a new connection for the data source

5. On the next screen, we need to select **Self-Hosted** and click on **Continue** (you can go through [Integration runtime in Azure Data Factory]() if you want to explore integration runtimes in more detail):



Integration runtime setup

Network environment:

Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:

Azure

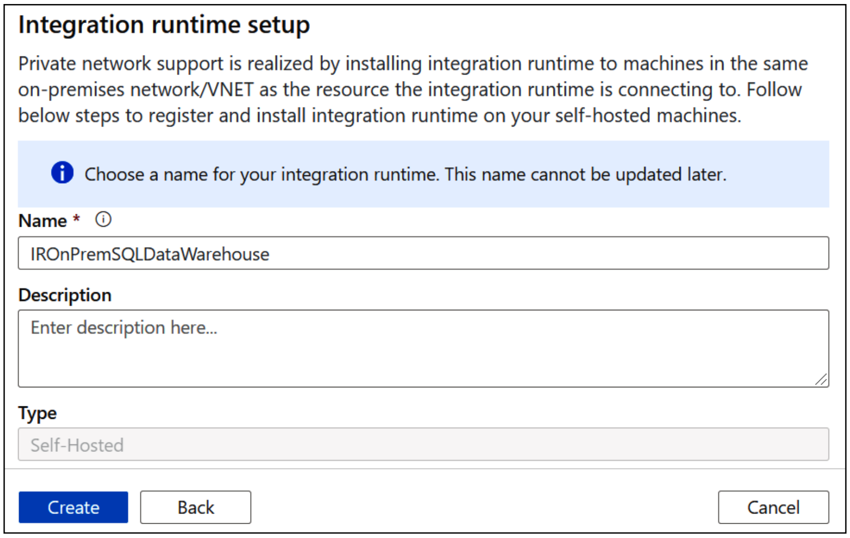
Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.

Self-Hosted

Use this for running activities in an on-premises / private network

Figure 13: Selecting the network environment for Integration runtime setup

6. Fill in the **Name** and **Description** fields for your integration runtime and then click on **Create**:



Integration runtime setup

Private network support is realized by installing integration runtime to machines in the same on-premises network/VNET as the resource the integration runtime is connecting to. Follow below steps to register and install integration runtime on your self-hosted machines.

Choose a name for your integration runtime. This name cannot be updated later.

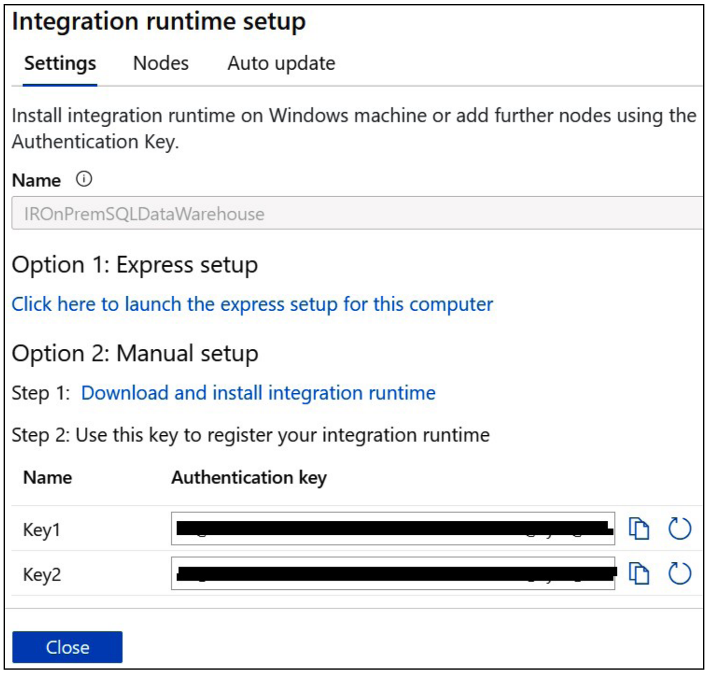
Name * : IROnPremSQLDataWarehouse

Description: Enter description here...

Type: Self-Hosted

Figure 14: Providing the name for the integration runtime

7. Now, click on the link right below **Option 1** to launch the express setup of the self-hosted integration runtime. Alternatively, you can also perform a manual setup by following the steps provided in **Option 2**:



Integration runtime setup

Install integration runtime on Windows machine or add further nodes using the Authentication Key.

Name: IROnPremSQLDataWarehouse

Option 1: Express setup

Click here to launch the express setup for this computer

Option 2: Manual setup

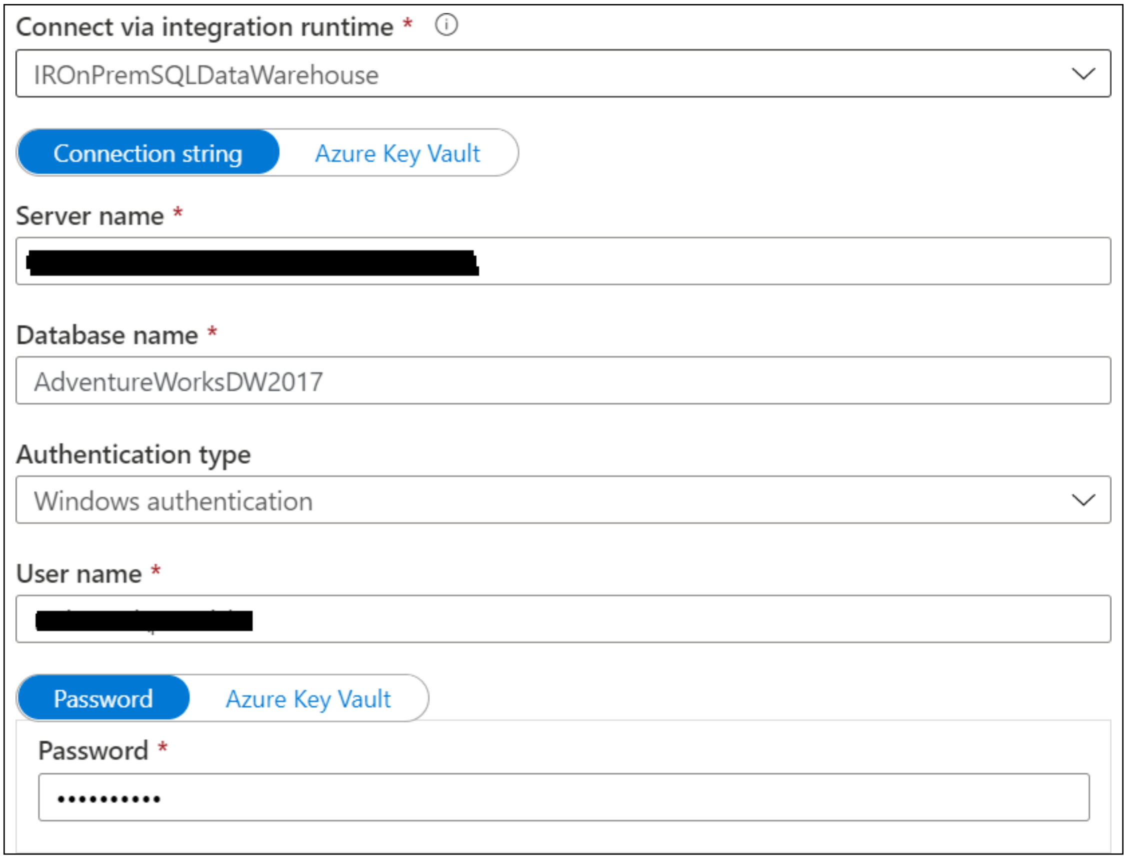
Step 1: Download and install integration runtime

Step 2: Use this key to register your integration runtime

Name	Authentication key
Key1	[redacted]
Key2	[redacted]

Figure 15: Setting up the integration runtime on the machine

8. Once the setup is complete, you can close this window and select the new integration runtime that you created just now from the **Connect via integration runtime** drop-down list.
9. Complete the corresponding **Server name** and **Database name** fields.
10. Select **Windows authentication** for **Authentication type**. Alternatively, you could also select **SQL authentication**.
11. Complete the **User name** and **Password** fields, click on **Test connection**, and then click on **Create**:



Connect via integration runtime *
IROnPremSQLDataWarehouse

Server name *
[redacted]

Database name *
AdventureWorksDW2017

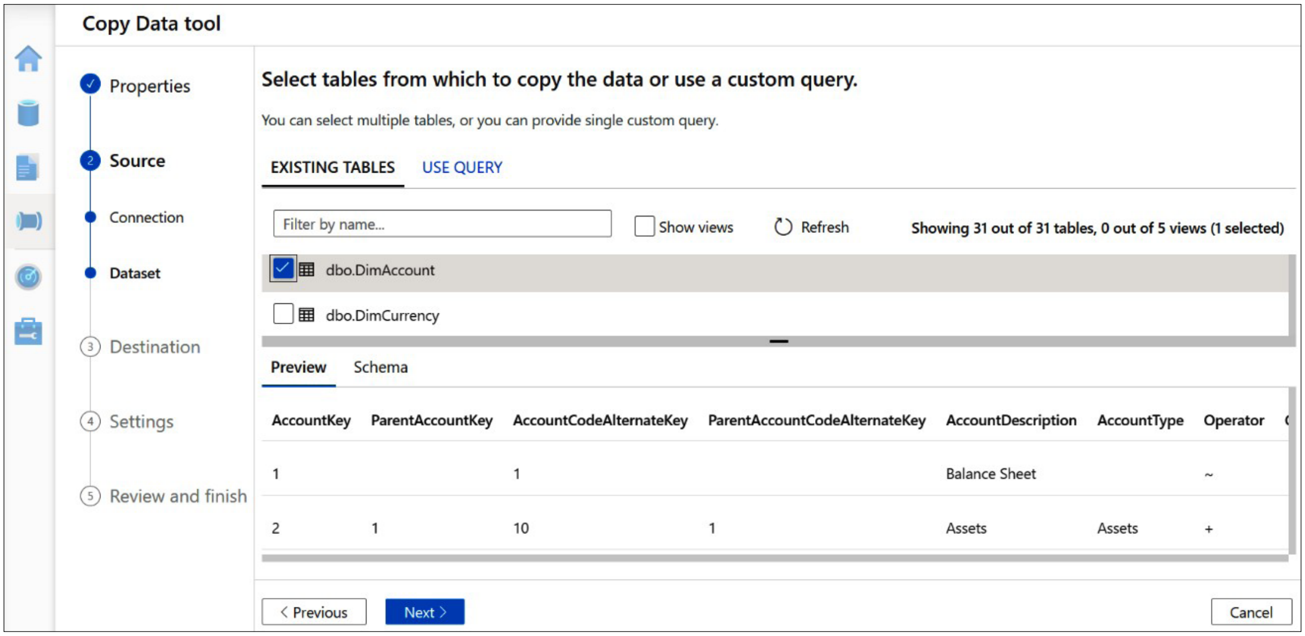
Authentication type
Windows authentication

User name *
[redacted]

Password *
••••••••••

Figure 16: Creating the connection for an on-premises SQL data warehouse

12. Now, click on **Next** to select a particular table we need to copy the data from. For now, let's select **DimAccount**, preview the records, and then click on **Next**:



Copy Data tool

Select tables from which to copy the data or use a custom query.

You can select multiple tables, or you can provide single custom query.

Showing 31 out of 31 tables, 0 out of 5 views (1 selected)

☑ dbo.DimAccount

□ dbo.DimCurrency

AccountKey	ParentAccountKey	AccountCodeAlternateKey	ParentAccountCodeAlternateKey	AccountDescription	AccountType	Operator
1		1		Balance Sheet		~
2	1	10	1	Assets	Assets	+

Figure 17: A snapshot of the preview of records for the selected table

13. Then, we need to define **Destination data source**. To do that, click on **+ Create new connection**.

14. Select **Azure Synapse Analytics** from the list of all available data sources and click on **Continue**:

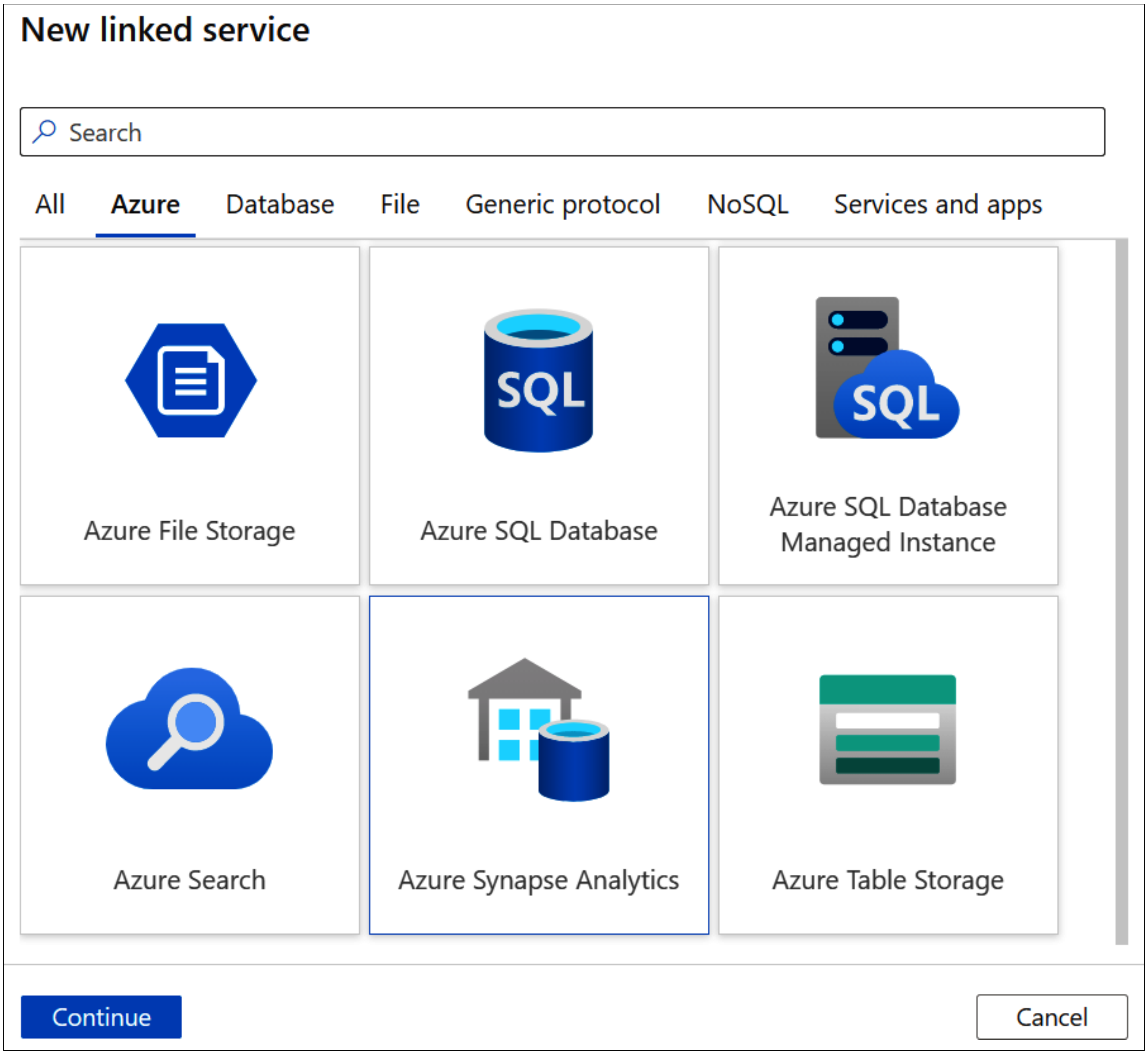
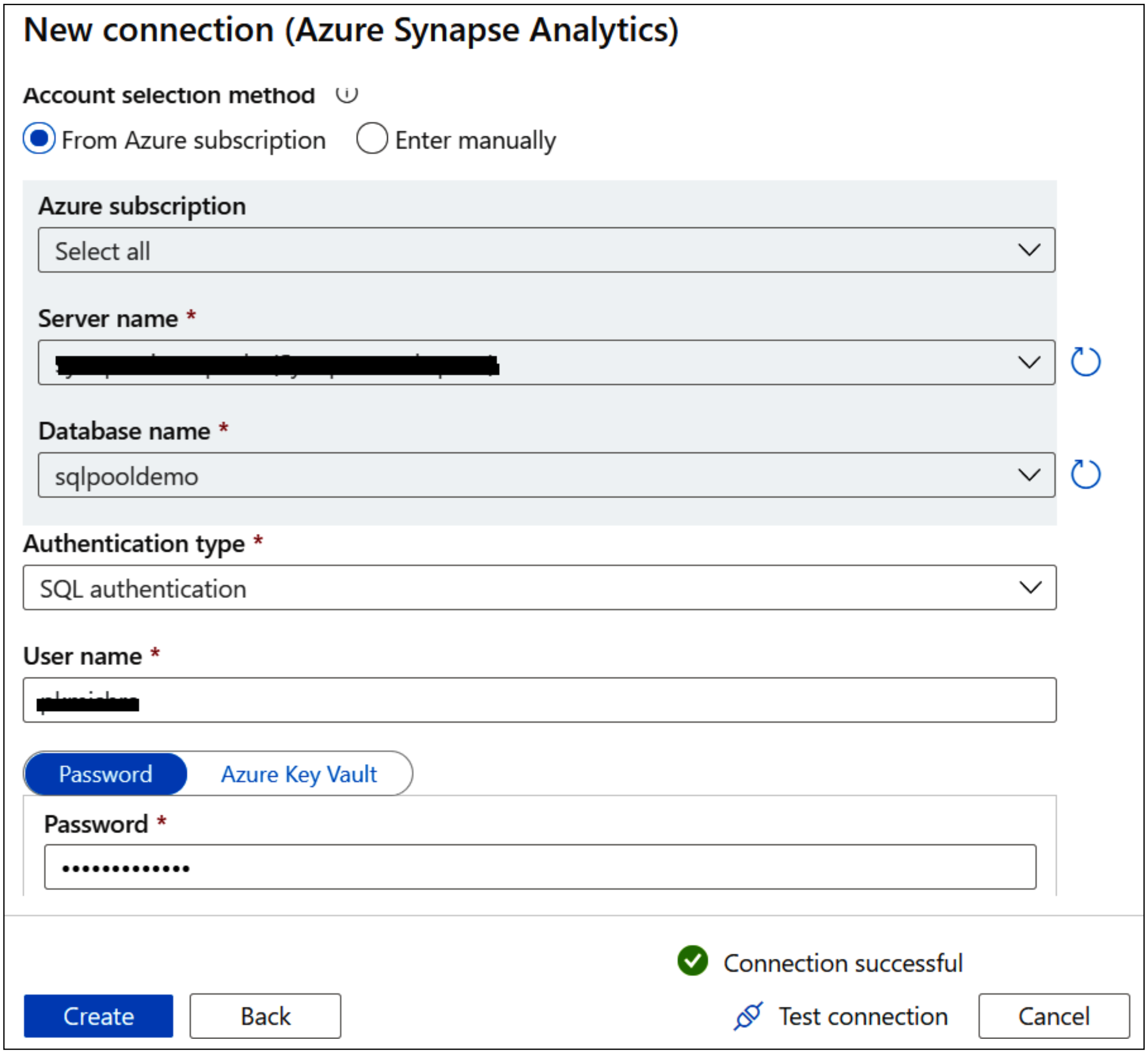


Figure 18: Selecting Azure Synapse Analytics to create a new linked service

15. Select the server name for your SQL pool, which is nothing other than your Synapse workspace name, and then select the database name, which is your SQL pool.

16. Select **SQL authentication** for **Authentication type** and complete the **User name** and **Password** fields to create the connection:



New connection (Azure Synapse Analytics)

Account selection method

● From Azure subscription ○ Enter manually

Azure subscription: Select all

Server name *: [redacted]

Database name *: sqlpooldemo

Authentication type *: SQL authentication

User name *: [redacted]

Password | Azure Key Vault

Password *: ••••••••••••••

Connection successful

Figure 19: Providing the details to create a new connection for the dedicated SQL pool

17. And finally, select the target table from the drop-down list of all the **Destination** tables to be mapped with the corresponding **Source** table. In this case, it should be **dbo.DimAccount**:

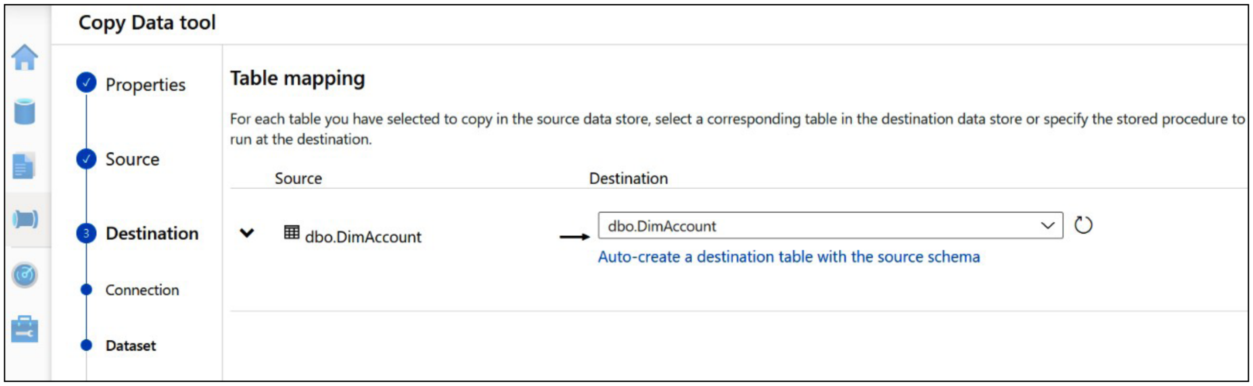


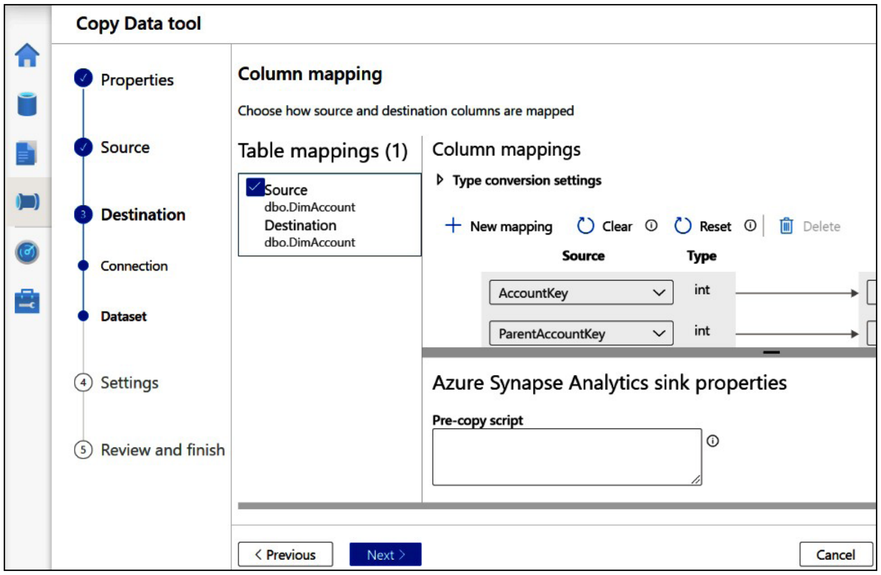
Table mapping

For each table you have selected to copy in the source data store, select a corresponding table in the destination data store or specify the stored procedure to run at the destination.

Source	Destination
dbo.DimAccount	dbo.DimAccount Auto-create a destination table with the source schema

Figure 20: A snapshot of Table mapping between source and target connections

18. Click on **Next** and review **Column mapping** for the selected **Source** and **Target** tables. You can change or modify some mappings as per your business requirements:



Column mapping

Choose how source and destination columns are mapped

Table mappings (1)

☑ Source dbo.DimAccount Destination dbo.DimAccount

Column mappings

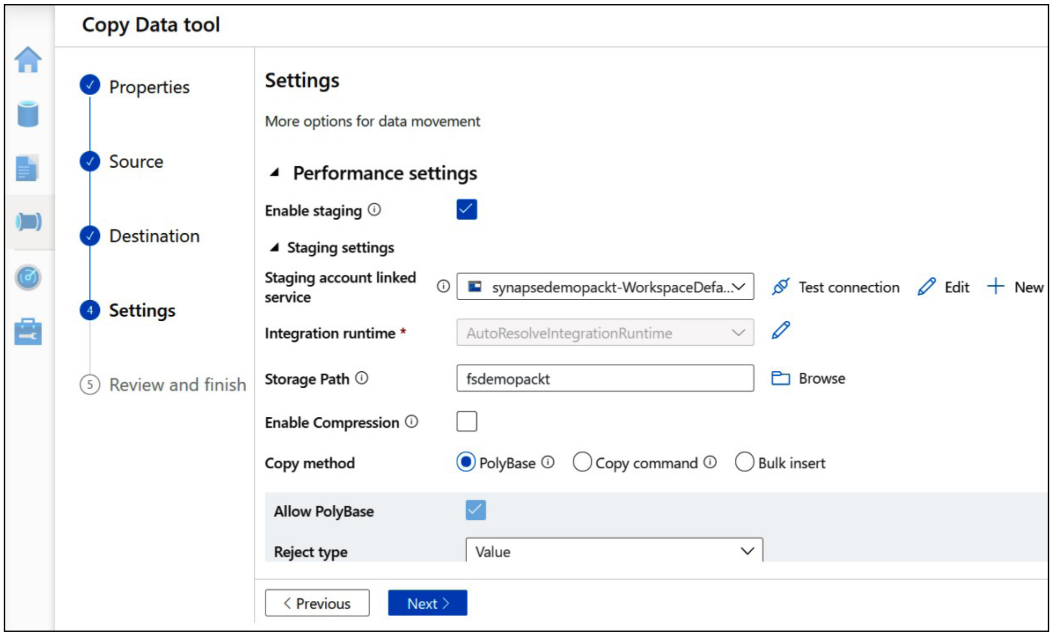
Source	Type
AccountKey	int
ParentAccountKey	int

Azure Synapse Analytics sink properties

Pre-copy script

Figure 21: Column mapping for the selected source and target tables

19. On the **Settings** page, tick the checkbox next to **Enable staging**, provide a **Storage Path** by selecting the **Browse** button next to it, and select **PolyBase** for **Copy method**. You are free to use other copy methods as well:



Copy Data tool

- ✓ Properties
- ✓ Source
- ✓ Destination
- 4 Settings
- 5 Review and finish

Settings

More options for data movement

Performance settings

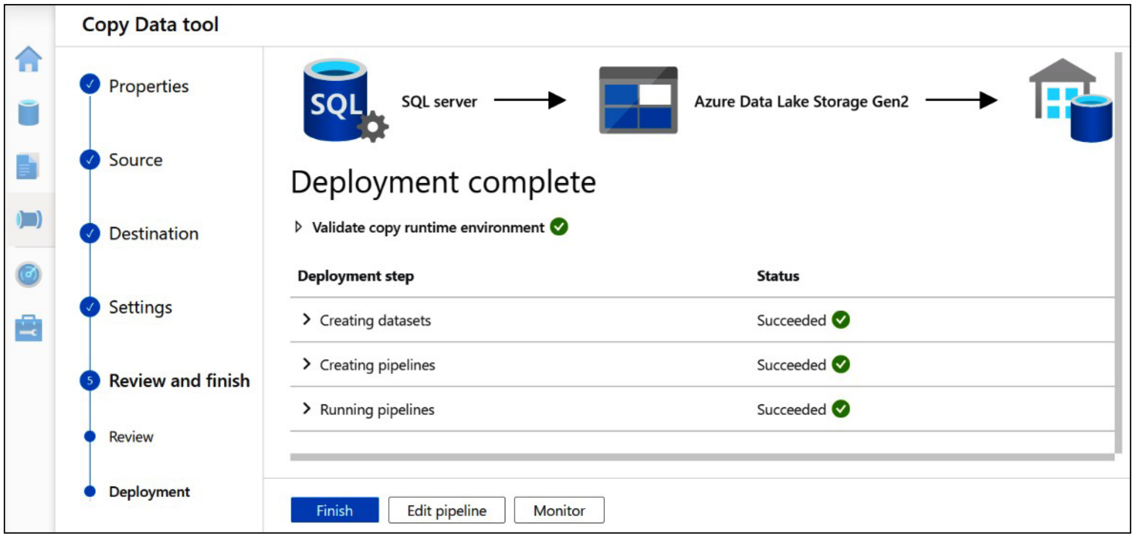
Enable staging ☑

Staging settings

Staging account linked service	synapsedemopackt-WorkspaceDefa...
Integration runtime *	AutoResolveIntegrationRuntime
Storage Path	fsdemopackt
Enable Compression	□
Copy method	● PolyBase ○ Copy command ○ Bulk insert
Allow PolyBase	☑
Reject type	Value

Figure 22: A snapshot of performance settings in the Copy Data tool

20. Leave default values for all the other settings. Click on **Next** to review the **Summary** screen and click on **Next** again to complete the pipeline. You can click on **Monitor** once all the deployment steps have been completed successfully:



Copy Data tool

- ✓ Properties
- ✓ Source
- ✓ Destination
- ✓ Settings
- 5 Review and finish
  - Review
  - Deployment

Deployment complete

Validate copy runtime environment ✓

Deployment step	Status
Creating datasets	Succeeded ✓
Creating pipelines	Succeeded ✓
Running pipelines	Succeeded ✓

Figure 23: A snapshot of the completed pipeline deployment

Similarly, you can use different tools and services to copy the data from an on-premises SQL data warehouse to an Azure Synapse dedicated SQL pool:

- **Synapse Spark pool**: Synapse Spark can also be used to create Spark notebooks for ETL operations. You can use SQL connectors for Apache Spark for accessing the data on your on-premises SQL data marts. You can get all the details about this connector by referring to Azure SQL Database and SQL Server connector for Apache Spark.

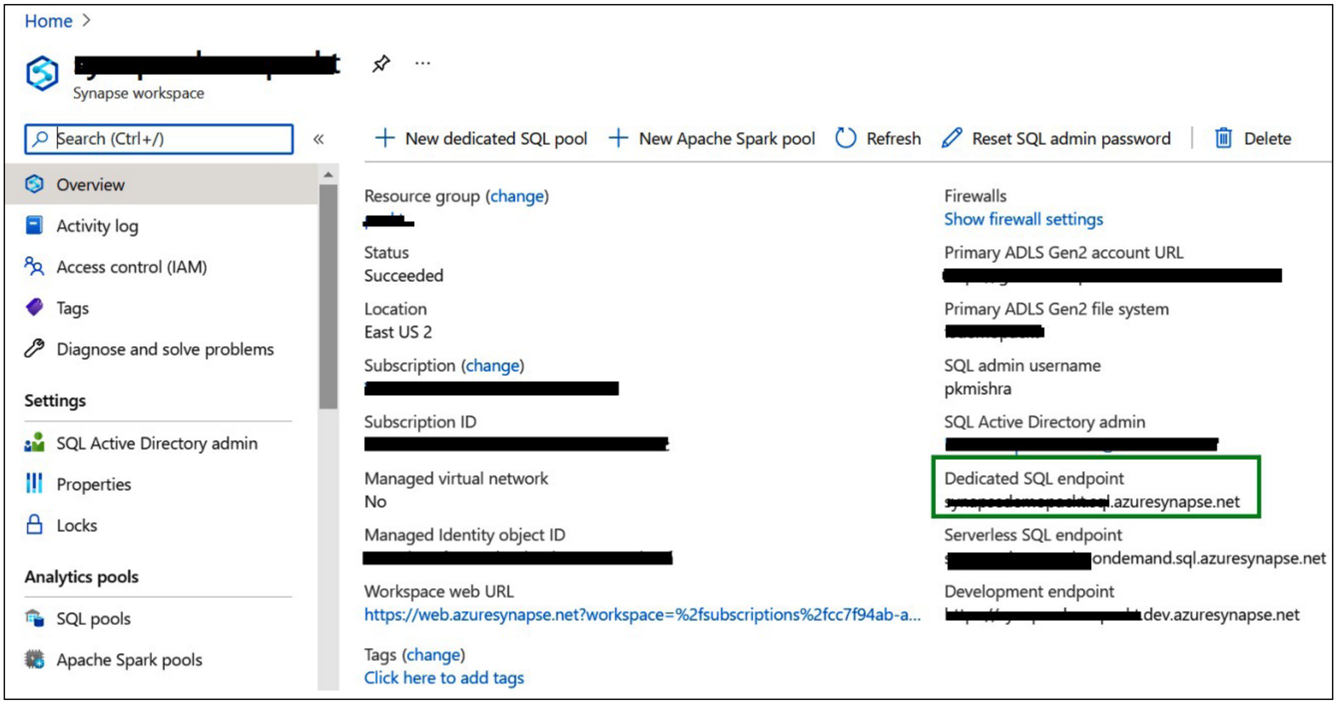
- **SSIS packages**: An **SSIS** (**SQL Server Integration Service**) package is a collection of various components required to execute a specific ETL task. Users can connect to various data sources and perform business transformations before feeding that data to the destination data storage. You can create these packages as well to copy the data from your on-premises SQL data warehouse to Synapse SQL pools. Step-by-step guidelines are provided at the following link: Load data into Azure Synapse Analytics with SQL Server Integration Services (SSIS) - SQL Server Integration Services (SSIS) | Microsoft Docs.

After building your new data warehouse, you need to decide what you want to do with it. With Azure Synapse, you have ample options to use the data in a direction that is apt for your business. Let’s learn about some of these options in the following section.

Step 4: Optimizing data warehouse workloads for BI and reporting

Like your traditional data warehouse, you can create a tabular model on top of your data in a SQL pool. However, now you can achieve the same solution by creating data modeling in Power BI. You can also create your star schema in a Power BI model instead of creating a multi-dimensional model in SQL Server Analysis Services (on-premises) and building reports on top of this data. To explore this in more detail, refer to [Understand star schema and the importance for Power BI](#).

Power BI Desktop can be connected to a Synapse SQL pool using the public endpoint that you will find on the **Overview** page of your Synapse workspace on the Azure portal, as highlighted in *Figure 24*:



Resource group (change)
[redacted]

Status
Succeeded

Location
East US 2

Subscription (change)
[redacted]

Subscription ID
[redacted]

Managed virtual network
No

Managed Identity object ID
[redacted]

Workspace web URL
https://web.azuresynapse.net?workspace=%2fsubscriptions%2fcc7f94ab-a...

Tags (change)
Click here to add tags

Firewalls
Show firewall settings

Primary ADLS Gen2 account URL
[redacted]

Primary ADLS Gen2 file system
[redacted]

SQL admin username
pkmishra

SQL Active Directory admin
[redacted]

Dedicated SQL endpoint
[redacted].azuresynapse.net

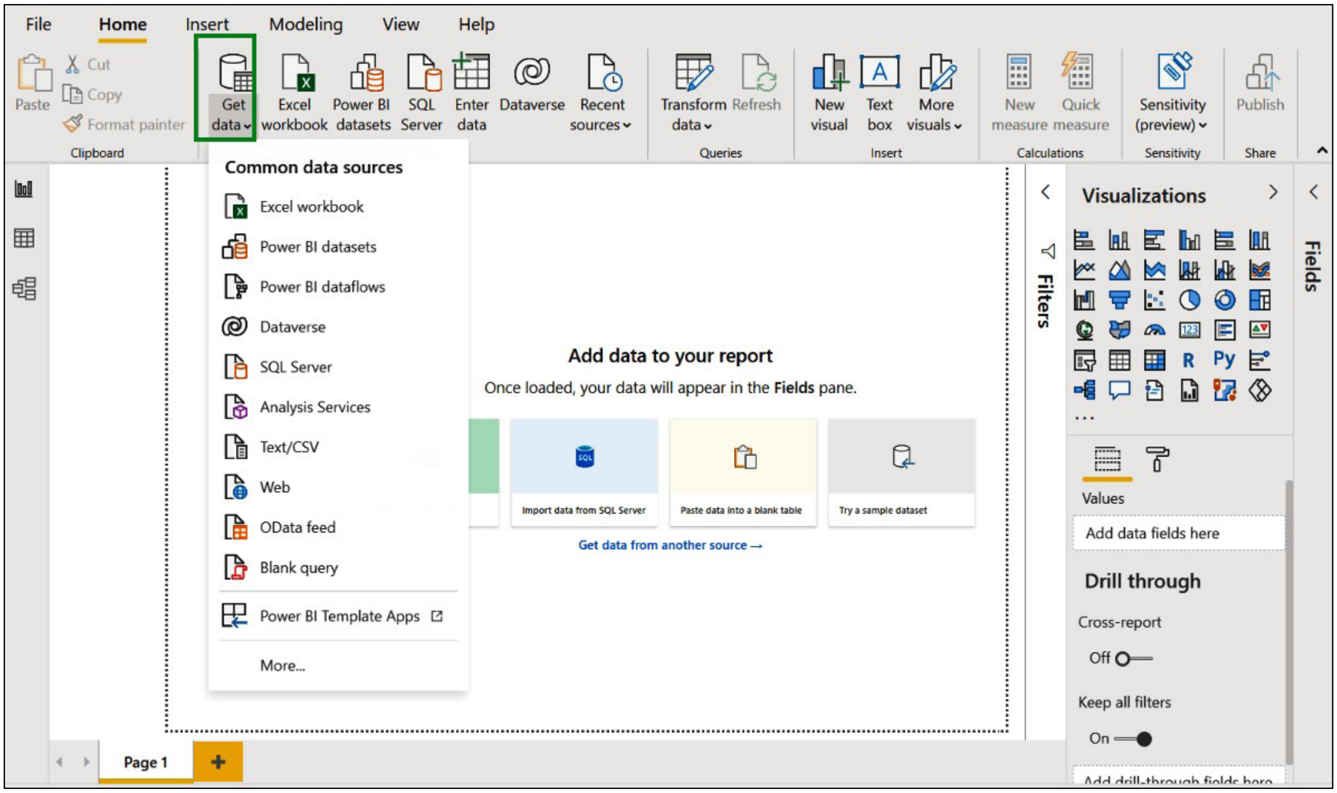
Serverless SQL endpoint
[redacted]ondemand.sql.azuresynapse.net

Development endpoint
[redacted]dev.azuresynapse.net

Figure 24: A snapshot of the Overview page of the Synapse workspace on the Azure portal

You can go through the following steps to connect your Power BI desktop to the Synapse SQL pool:

1. Open the **Power BI Desktop** tool on your machine and click on **Get data**, as highlighted in *Figure 25*:



Add data to your report

Once loaded, your data will appear in the **Fields** pane.

Figure 25: A snapshot of Power BI Desktop

2. Click on **More...** at the bottom of this drop-down list to get the list of all the supported data sources, select **Azure Synapse Analytics (SQL DW)**, and then click on **Connect**:

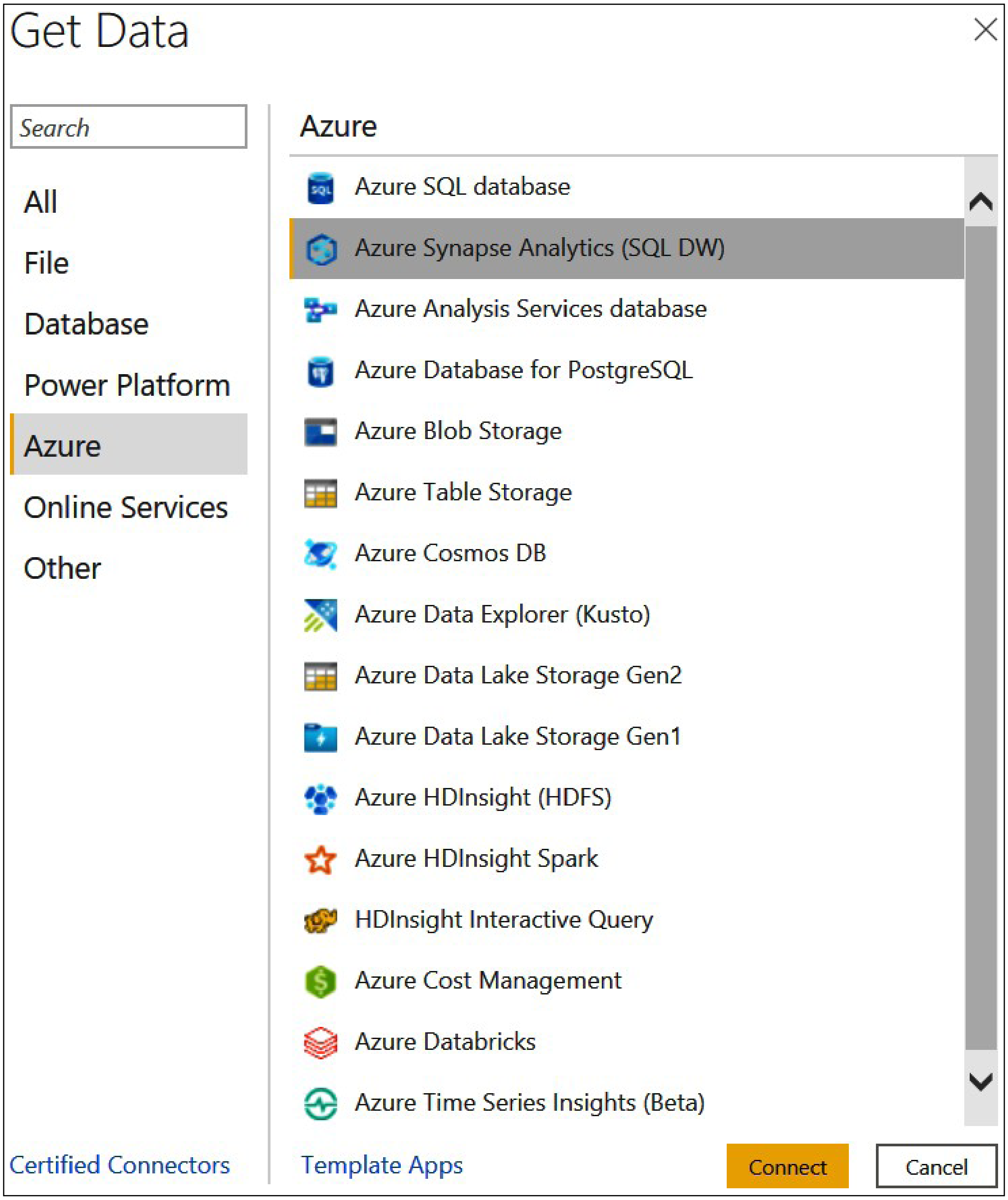
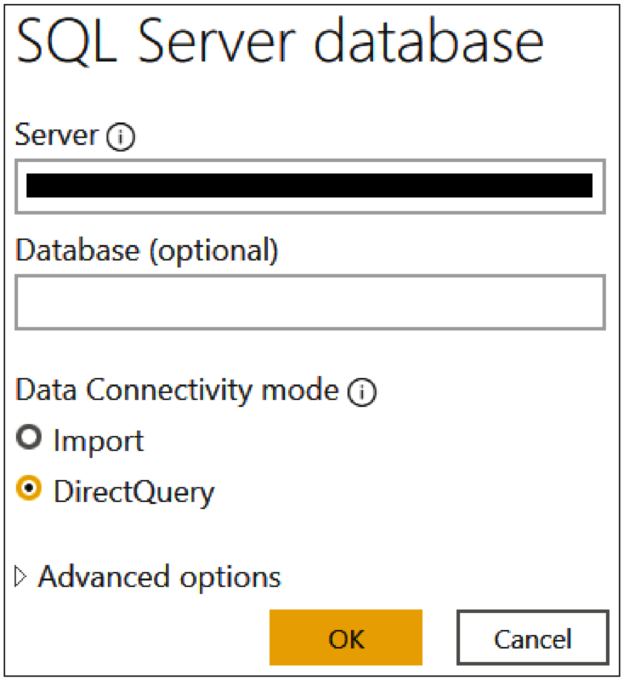


Figure 26: Selecting the data source for bringing the data to Power BI

3. Copy the **Dedicated SQL endpoint** field from the **Overview** page of your Synapse workspace and paste it in the **Server** field on Power BI Desktop.
4. Select **DirectQuery** for **Data Connectivity mode** and then click on **OK**:



SQL Server database

Server ⓘ

[redacted]

Database (optional)

Data Connectivity mode ⓘ

○ Import

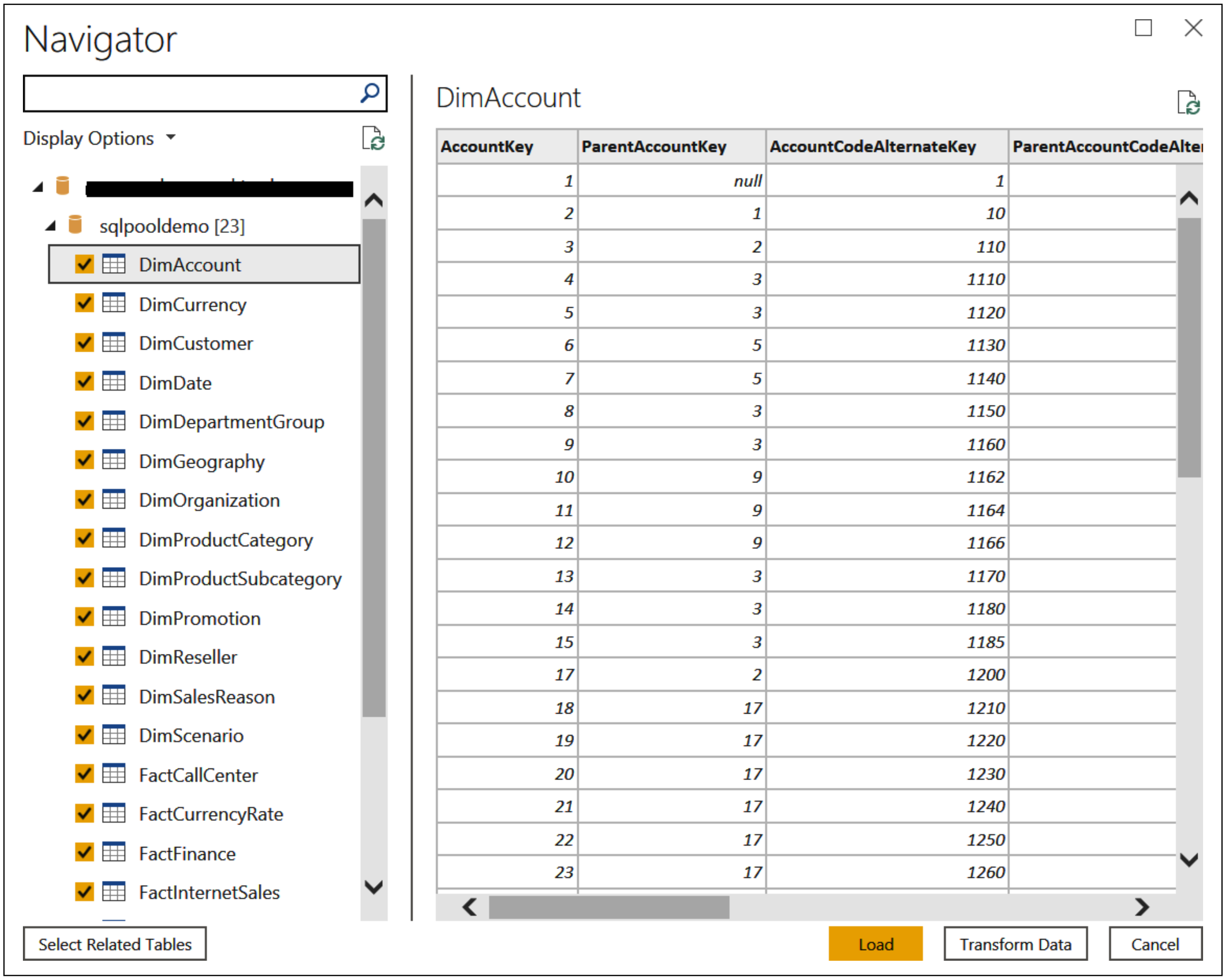
● DirectQuery

▷ Advanced options

Figure 27: Providing the server details for the SQL pool in Power BI

5. Complete the **User name** and **Password** fields for your **Database** authentication and click on **OK**.

6. Select your data warehouse and required table that you would like to load to the Power BI data model and then click on **Load**:



Navigator

Display Options ▾

◢ [redacted]

◢ sqlpooldemo [23]

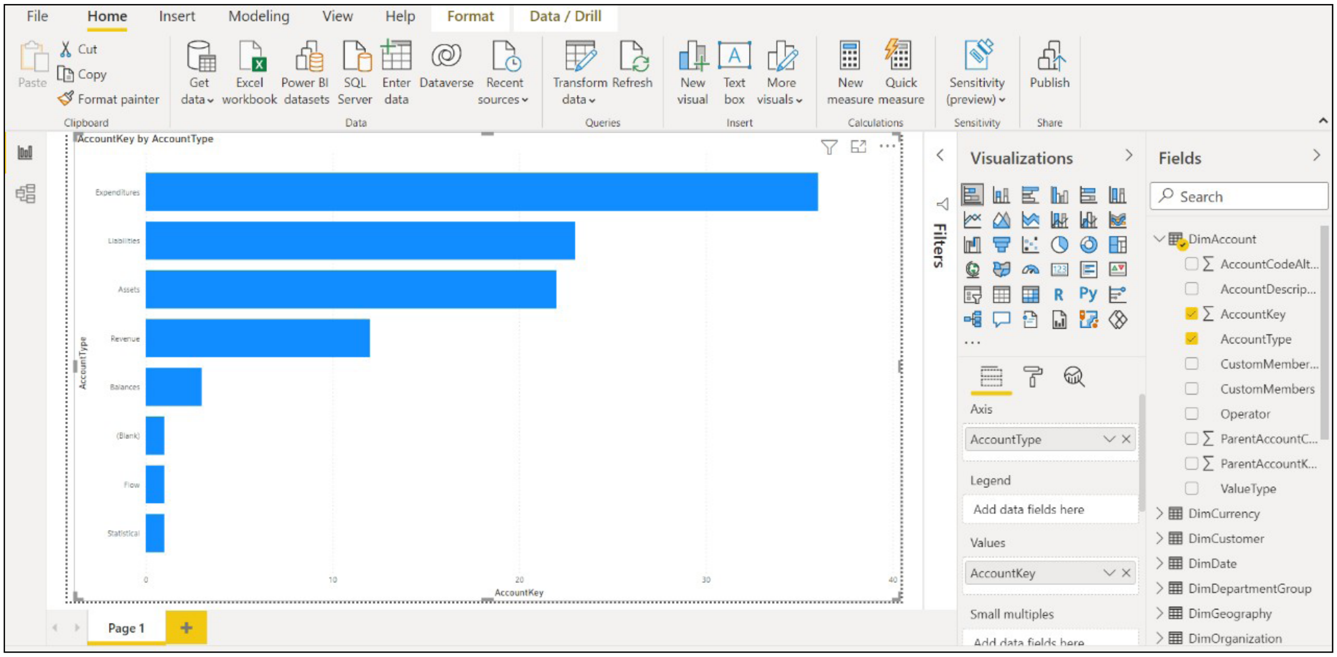
☑ DimAccount
☑ DimCurrency
☑ DimCustomer
☑ DimDate
☑ DimDepartmentGroup
☑ DimGeography
☑ DimOrganization
☑ DimProductCategory
☑ DimProductSubcategory
☑ DimPromotion
☑ DimReseller
☑ DimSalesReason
☑ DimScenario
☑ FactCallCenter
☑ FactCurrencyRate
☑ FactFinance
☑ FactInternetSales

DimAccount

AccountKey	ParentAccountKey	AccountCodeAlternateKey	ParentAccountCodeAlte
1	null	1	
2	1	10	
3	2	110	
4	3	1110	
5	3	1120	
6	5	1130	
7	5	1140	
8	3	1150	
9	3	1160	
10	9	1162	
11	9	1164	
12	9	1166	
13	3	1170	
14	3	1180	
15	3	1185	
17	2	1200	
18	17	1210	
19	17	1220	
20	17	1230	
21	17	1240	
22	17	1250	
23	17	1260	

Figure 28: Loading the data from different tables to Power BI

7. Now, all the tables and corresponding columns can be seen under the **Fields** pane, on the right side of your Power BI tool. Select **Stacked bar chart** from the list of all the visualizations available in the **Visualizations** pane and select **AccountType** for the **Axis** field and **AccountKey** for the **Values** field for the chart:



Axis
AccountType

Legend
Add data fields here

Values
AccountKey

Small multiples
Add data fields here

DimAccount
- ☐ AccountCodeAlt...
- ☐ AccountDescrip...
- ☑ AccountKey
- ☑ AccountType
- ☐ CustomMember...
- ☐ CustomMembers
- ☐ Operator
- ☐ ParentAccountC...
- ☐ ParentAccountK...
- ☐ ValueType

DimCurrency
DimCustomer
DimDate
DimDepartmentGroup
DimGeography
DimOrganization

Figure 29: Generating reports on Power BI

8. Similarly, you can add many visualizations to your Power BI Desktop. Once you are done with all the reports, you can click on the **Publish** link in the top-right corner of your Power BI to publish the dashboard to the Power BI service.

These reports can be accessed directly on your Synapse Studio by creating a linked service for Power BI. You can also create new reports in Synapse Studio itself. Please go to Linking a Power BI workspace to a Synapse workspace to learn more about this feature.

To learn about modeling your data in Power BI, refer to Model data in Power BI.

Step 5: Cloud-scale machine learning for analytics

Now that you know how to optimize your data warehouse workloads, it's time to use ML services to perform analytics operations on the data.

Users can build powerful, cloud-based machine learning applications using Azure Machine Learning services and now you can use this service in conjunction with Azure Synapse. In this section, we'll perform analytics on operational data with Azure Synapse Link and explore new Azure Synapse features to integrate predictive analytics capabilities.

Big data and operational analytics

In this section, we will cover some brief concepts surrounding the Apache Spark pool and how you can perform analytics on operational data with Azure Synapse Link.

Spark pool

A Spark pool is Microsoft's implementation of the **Apache Spark** engine in the cloud. It enables you to use the Spark engine without worrying about the infrastructure overhead of managing the Spark node. A Spark pool can be used to process data on **Azure Storage** as well as **Azure Data Lake Gen2 Storage**. A Spark pool comes with many libraries built on top of Spark that can be used for data processing and connectivity. Please refer to Analyze with Apache Spark to learn more about the Synapse Spark pool.

Synapse link for Cosmos DB

A **hybrid transactional and analytical processing** (**HTAP**) capability enables you to run near real-time analytics over operational data in Cosmos DB. It uses two different storage layers of Cosmos DB, a **transactional store** and an **analytical store**. Users need to enable the analytical store while creating a new container on Cosmos DB. Both the stores are kept in sync automatically. Azure Synapse is integrated with the analytical store of Cosmos DB via the Synapse link:

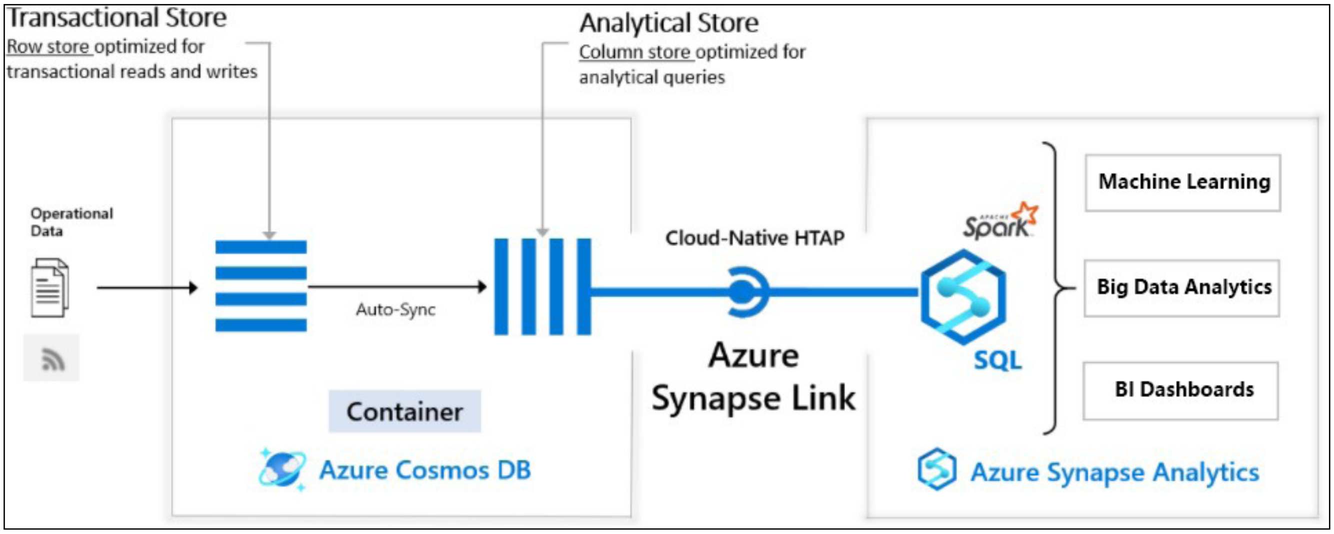


Figure 30: Components of Azure Synapse Link for Cosmos DB

You can use OPENROWSET syntax to analyze the data in the analytical store of the Azure Cosmos DB container:

```
OPENROWSET(
        ‘CosmosDB’,
        ‘<Azure Cosmos DB connection string>’,
        <Container name>
    ) [ < with clause > ]
```

You can read more about Azure Synapse Link for Cosmos DB in the following documentation: [What is Azure Synapse Link for Azure Cosmos DB?]

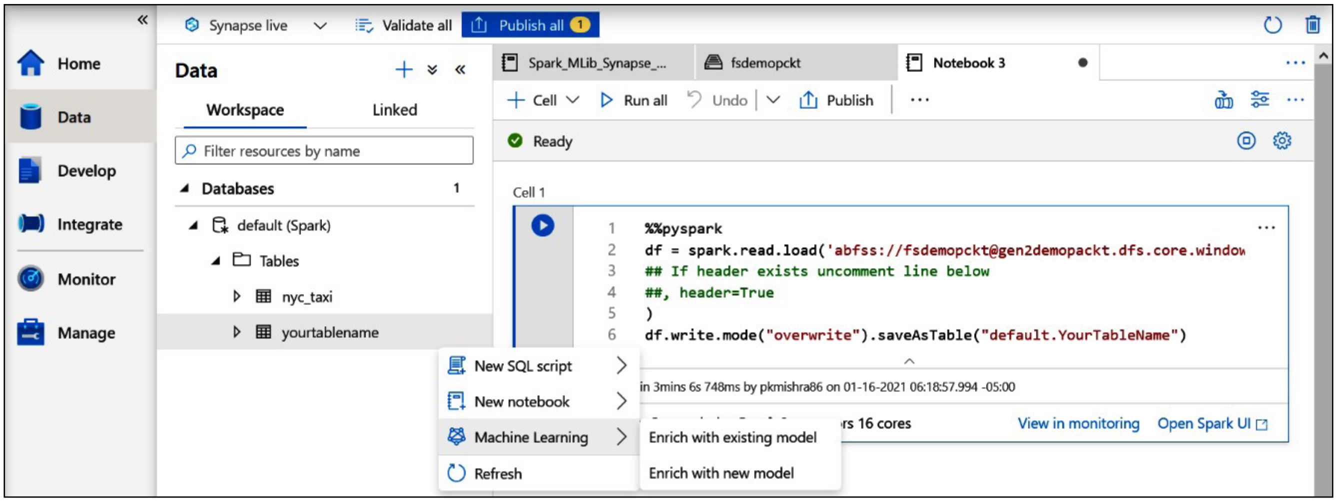
Data science and predictive analytics

In this section, we will discover new Azure Synapse features to integrate predictive analytics capabilities with Azure Synapse. You can refer to Prerequisites for using Cognitive Services in Azure Synapse Analytics to learn how to configure cognitive services with Azure Synapse.

Azure Synapse enables you to easily enrich your data in Azure Synapse with existing models of cognitive services. At the time of writing, you can see two existing models to enrich your data, **sentiment analysis** and **Anomaly Detector**. We will learn about both these options in the following sections.

You can perform sentiment analysis on your text data with the existing models available on Azure Synapse. But first, you need to have your data loaded to a Spark table. Make sure your file is uploaded to the Azure Data Lake Gen2 account, which is configured as the default storage for your Azure Synapse workspace. You need to make sure that you have the contributor level permission on the Azure Data Lake Gen2 filesystem where your data resides:

1. Go to the **Data** tab in Azure Synapse Studio and expand your **default (Spark)** database.
2. Right-click on your Spark table, select **Machine Learning** from the drop-down list, and click on **Enrich with existing model**:

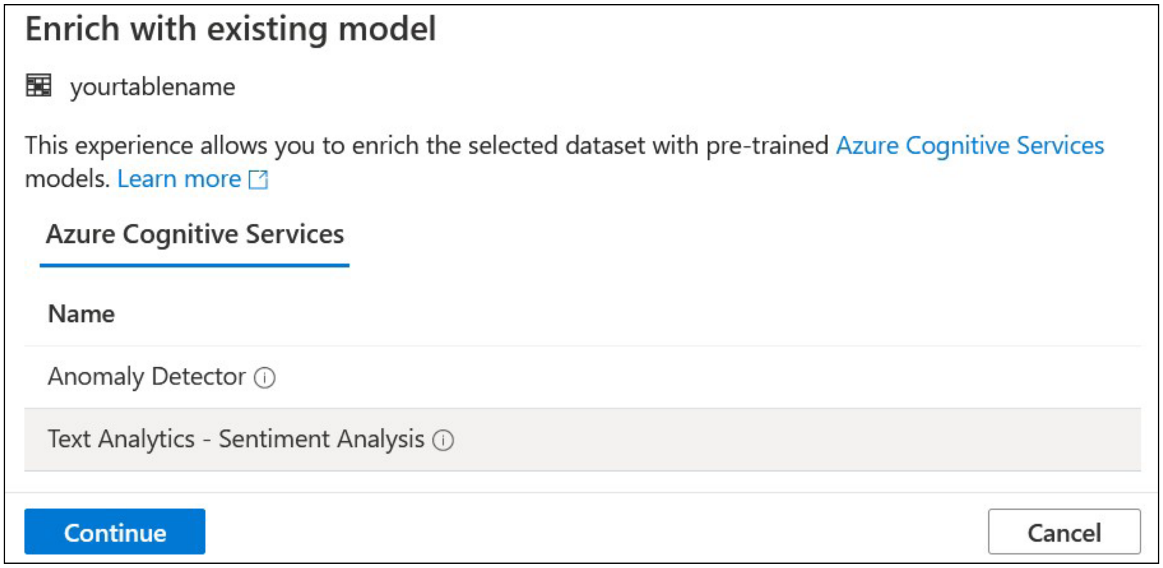


```
%%pyspark
df = spark.read.load('abfss://fsdemopckt@gen2demopackt.dfs.core.window
## If header exists uncomment line below
##, header=True
)
df.write.mode("overwrite").saveAsTable("default.YourTableName")
```

```
in 3mins 6s 748ms by pkmishra86 on 01-16-2021 06:18:57.994 -05:00
```

Figure 31: A snapshot of enriching your data in a Spark table

3. Select **Text Analytics - Sentiment Analysis** from the list of existing models and click on **Continue**. You can also select **Anomaly Detector** as per your business requirements:



Enrich with existing model

yourtablename

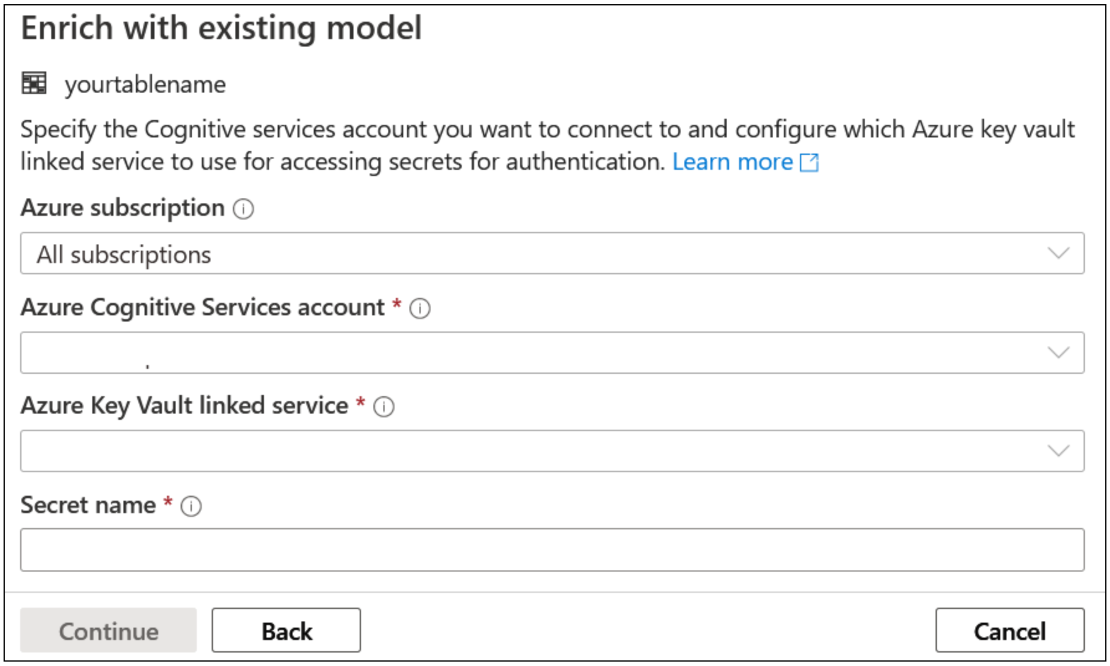
This experience allows you to enrich the selected dataset with pre-trained Azure Cognitive Services models. Learn more

Azure Cognitive Services

Name
Anomaly Detector ⓘ
Text Analytics - Sentiment Analysis ⓘ

Figure 32: Selecting Text Analytics - Sentiment Analysis for enriching data in Azure Synapse

4. Provide the details for **Azure Cognitive Services account** and **Azure Key Vault linked service**:



Enrich with existing model

yourtablename

Specify the Cognitive services account you want to connect to and configure which Azure key vault linked service to use for accessing secrets for authentication. Learn more

Azure subscription ⓘ: All subscriptions

Azure Cognitive Services account * ⓘ:

Azure Key Vault linked service * ⓘ:

Secret name * ⓘ:

Figure 33: Providing configuration for your Cognitive Services

5. Next, we just need to select the **Language** and **Text** columns that will be used for sentiment analysis and click on **Open notebook**.
6. Now, click on **Run All** to run all the cells on the notebook and you can see the result.

Now we have learned how we can use existing ML models to enrich our data with just a few steps. It is not possible to cover all the topics in relation to Azure ML services and their integration with Azure Synapse in this e-book, but we have covered most of the important topics that will help you explore the benefits and usage of Azure ML services integration with Azure Synapse.

You can refer to the following links to learn more about implementing Apache Spark ML with Azure Synapse:

- Machine learning with Apache Spark
- Build a machine learning app with Apache Spark MLlib and Azure Synapse Analytics

Now you are well equipped to modernize your data warehouse using the tools and techniques mentioned in this book. Azure Synapse also provides some additional features that you can leverage to get the best of Azure Synapse.

Integrate data analytics and data governance

Post migration to Azure Synapse, it is important to continue governing and managing your data estate to better enable data discovery and comply with an ever growing list of regulatory requirements. Azure Purview as a unified data governance solution helps you answer important questions like what data do I have? Where did that data originate? And can I trust that data?

Through Azure Purview's turnkey integration with Azure Synapse, you can easily and quickly link your Azure Purview account to an Azure Synapse workspace. This connection allows you to discover Azure Purview data assets from within the Azure Synapse search bar, connect that data to your Azure Synapse workspace with linked services or integration datasets, analyze those datasets with Azure Synapse Apache Spark, Azure Synapse SQL, and Data Flows; execute pipelines, and push lineage information to Azure Purview.

Your migration journey ends with you being able to scan, inventory and annotate the assets that landed in Azure Synapse, while also validating the lineage of the data transfer.

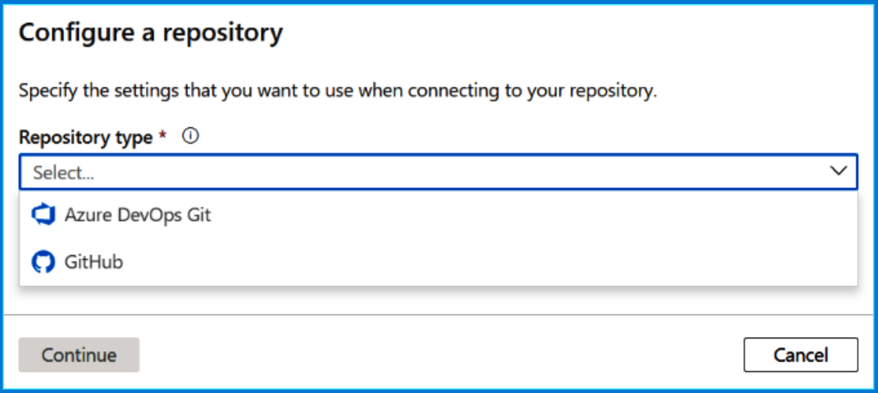
To learn more about how to connect an Azure Purview account to an Azure Synapse workspace, refer to the following link: Connect Synapse workspace to Azure Purview - Azure Synapse Analytics | Microsoft Docs

Additional resources

Although you have learned enough to modernize your data warehouse, it's always better to have a look at some of the additional resources and features offered by Azure Synapse.

Source control

Azure Synapse enables you to set up source control in Synapse Studio in order to collaborate with other team members. Synapse Studio includes built-in support for **Azure DevOps Git** and **GitHub** to configure the repository for your code:



Configure a repository

Specify the settings that you want to use when connecting to your repository.

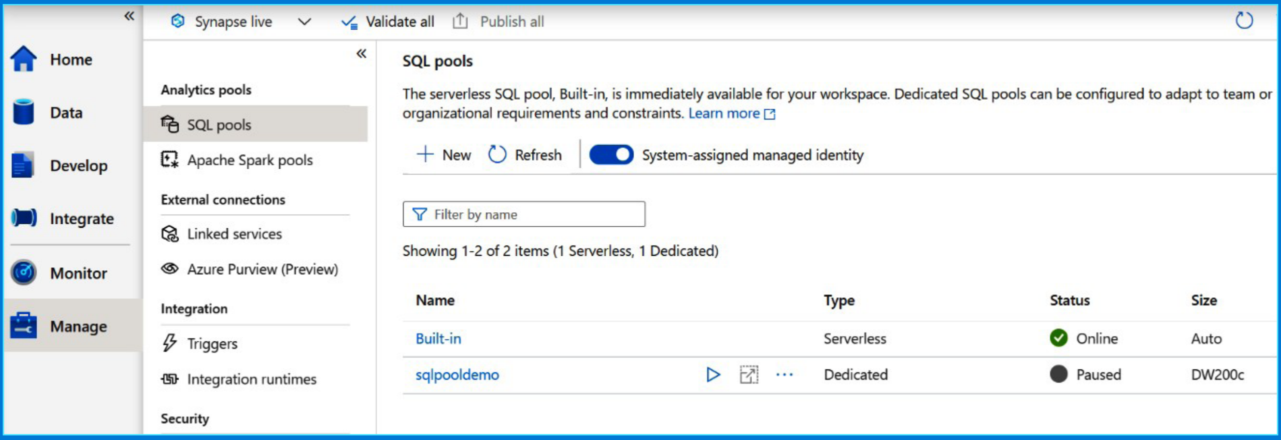
Figure 34: Configuring a repository in Azure Synapse

You can refer to Source control in Synapse Studio if you want to explore this feature.

Cost management

Cost management is one of the most important reasons why someone decides to move their workload to the cloud. Azure Synapse provides you with various ways to optimize your storage and operational costs. When you are running your data warehouse on-premises, you need to pay for your compute even when it's not in use. If you need Spark capabilities, you need to procure a machine to set up Spark so that it can be used whenever needed.

However, Azure Synapse enables you to pause or resume your compute engine as per business requirements without affecting storage:



SQL pools

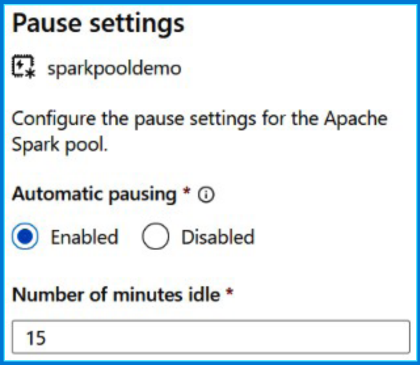
The serverless SQL pool, Built-in, is immediately available for your workspace. Dedicated SQL pools can be configured to adapt to team or organizational requirements and constraints. Learn more

Showing 1-2 of 2 items (1 Serverless, 1 Dedicated)

Name	Type	Status	Size
Built-in	Serverless	✓ Online	Auto
sqlpooldemo	Dedicated	● Paused	DW200c

Figure 35: Configuring a repository in Azure Synapse

Similarly, you can enable **Automatic pausing** for your Spark pool within the **Manage** hub of your Synapse Studio:



Pause settings

sparkpooldemo

Configure the pause settings for the Apache Spark pool.

Automatic pausing *

● Enabled ○ Disabled

Number of minutes idle *

15

Figure 36: Enabling Automatic pausing for the Spark pool

Although these are basic features of cost management, you can learn about more options for managing your costs on Azure Synapse by referring to Plan and manage costs for Azure Synapse Analytics.

Best practices

Users must follow the best practice guidelines to get the best out of any product. Azure Synapse also has its own set of best practices that will enable users to get the best results for their analytical processing.

You can refer to the following links to dive deeper into the corresponding best practice guidelines offered by Azure Synapse:

- Best practices for serverless SQL pool in Azure Synapse Analytics
- Best practices for dedicated SQL pools in Azure Synapse Analytics
- Best practices for loading data into a dedicated SQL pool in Azure Synapse Analytics
- Development best practices for Synapse SQL
- Optimize Apache Spark jobs in Azure Synapse Analytics

Get started today

Register for the free six-part hands-on training series for Azure Synapse Analytics.

Automate code translation and data migration from SQL Server data marts to Azure Synapse:

- Accelerator toolkit: GitHub – microsoft/AzureSynapseScriptsAndAccelerators
- Accelerator toolkit training video: SQL Server to Azure Synapse Analytics Migration Overview – YouTube

Start your migration journey today by accessing Azure Synapse Pathway from the Microsoft Download Center.

Are you a SQL Server professional? Learn how to extend your SQL Server skillsets in Azure Synapse.

Connect with an Azure specialist to learn more about bringing analytics in Azure to your organization.

This document is provided "as is." Information and views expressed in this document, including URL and other internet website references, may change without notice. You bear the risk of using it. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.