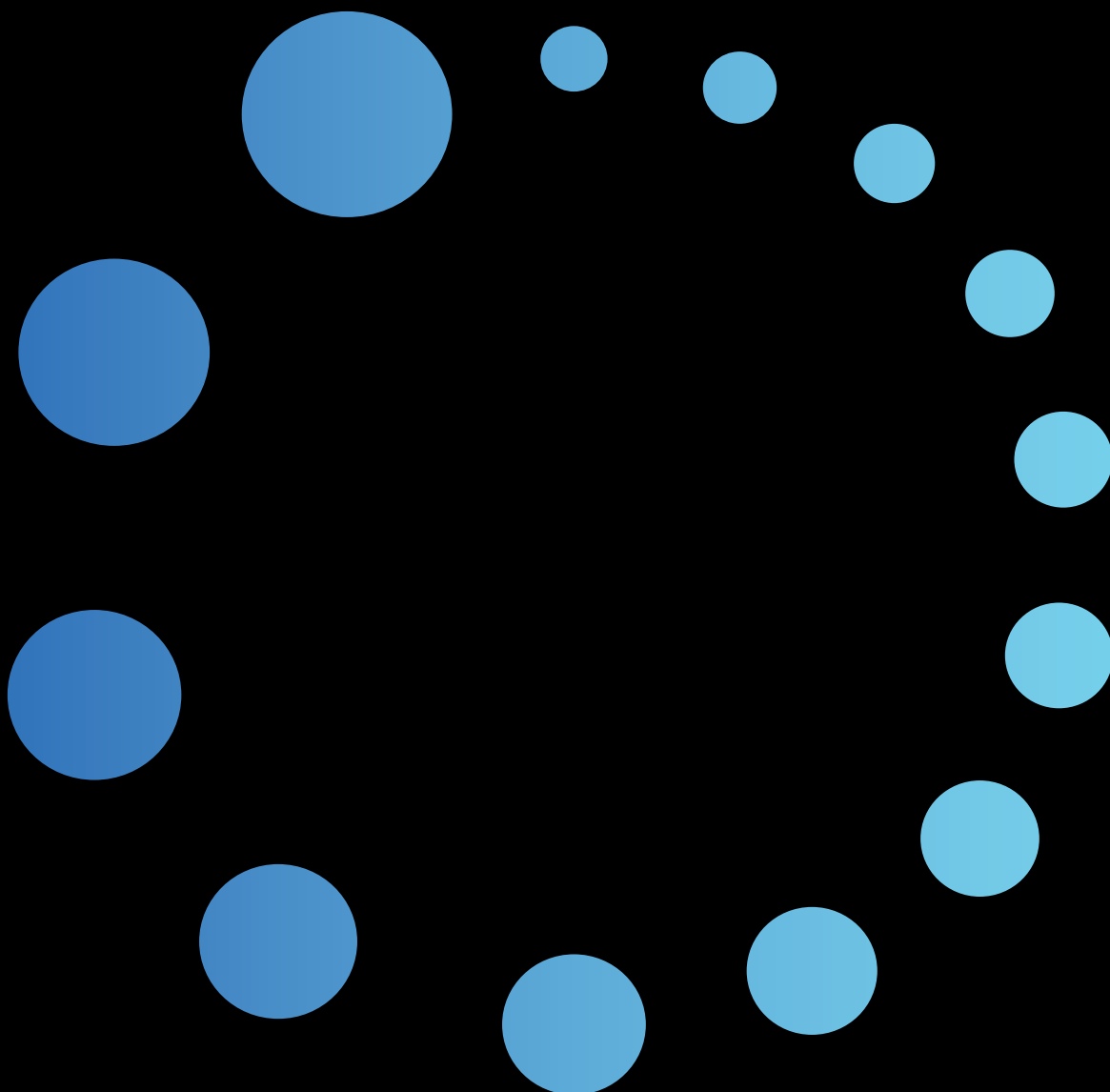


The Power BI Professional's Guide to Azure Synapse Analytics



Summary

This guide introduces Power BI practitioners to Azure Synapse Analytics—a limitless analytics service that brings together enterprise data warehousing and big data analytics.

On the surface, Azure Synapse Analytics is Azure SQL Data Warehouse evolved. However, it's much more than just a few new capabilities in an update of SQL Data Warehouse. Azure Synapse represents a modern, holistic, and unified approach to analytics that is unique in the industry. As an integrated cloud-native service encompassing previously isolated functions, such as data integration, data warehousing, and big data processing, Azure Synapse empowers Power BI professionals across a diverse set of use cases to deliver the scale, performance, and cost management their projects require.

This guide explores the deep integration of Power BI with Azure Synapse as both a data source and a development platform, and identifies the primary benefits of using Azure Synapse for new and existing solutions.

04 /

Introducing Azure Synapse Analytics

- 05 Azure Synapse SQL

06 /

Benefits of Azure Synapse for Power BI

- 06 Single source of truth
- 06 DirectQuery at scale
- 07 Centralized security
- 09 Team collaboration
- 10 Data preparation
- 10 Paginated report flexibility

11 /

Building Power BI solutions with Azure Synapse

- 11 Accessing an Azure Synapse workspace
- 13 Workspace vs. resource access
- 13 Connecting to Power BI in the Azure Synapse studio
- 15 Creating Power BI datasets via the Azure Synapse studio
- 17 Building reports in the Azure Synapse studio
- 20 Creating paginated reports
- 20 Power BI dataset vs. the SQL pool
- 21 Connecting to the SQL resource
- 24 Developing dataflows
- 27 AI predictive analytics integration
- 27 Composite models and aggregations
- 28 Targeted performance via aggregations
- 31 Table storage mode
- 32 Blending sources and connectivity

Introducing Azure Synapse Analytics

Azure Synapse is an end-to-end cloud-native analytics platform that brings together data ingestion, data warehousing, and big data into a single service. It gives you the freedom to query data on your terms, using either serverless or provisioned resources—at scale. The worlds of data warehousing and big data analytics come together in a unified experience ready to ingest, prepare, manage, and serve data for immediate BI and machine learning needs.

The Azure Synapse platform is integrated with linked services, including Power BI, Azure Machine Learning, and Azure Data Share. Interactive Power BI reports and enterprise-grade semantic models can be developed within the Azure Synapse studio, the new common web portal for developing and managing various Azure Synapse artifacts.

With the following architecture, Azure Synapse can ingest both structured and unstructured data and offers extract-transform-load (ETL), big data, and data warehousing technologies, all within a single unified service:

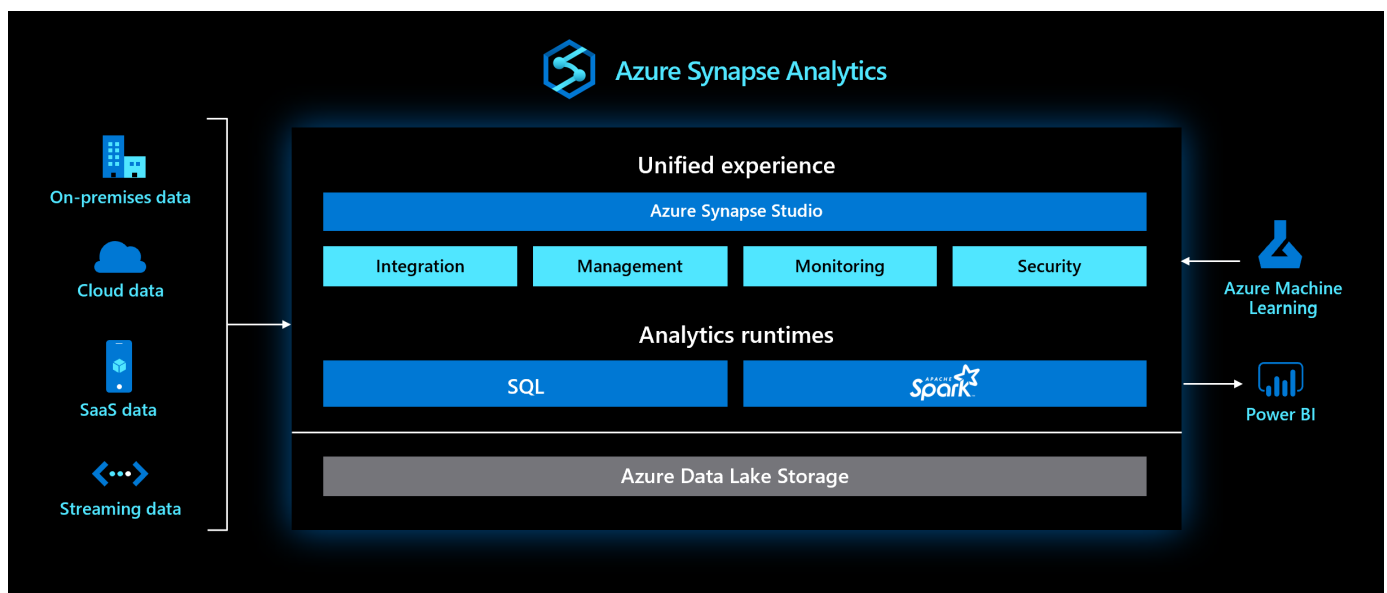


Figure 1: Azure Synapse Analytics

Azure Synapse SQL

Agility and rapid data exploration capabilities over large datasets in a data lake are highly valued features of modern data platforms. [Azure Synapse SQL](#) is the one-stop-shop for analyzing data using SQL technology.

Synapse SQL gives you the freedom to query data using the following two form factors:

- Provisioned data warehouse with SQL pools
- Serverless queries over the data lake

To address the need for on-demand computing power, Synapse SQL offers data engineers the ability to run serverless queries without having to provision any infrastructure.

In the following image from the Azure Synapse studio, the serverless endpoint is used to execute a query against a collection of Parquet files stored in Azure Data Lake Storage:

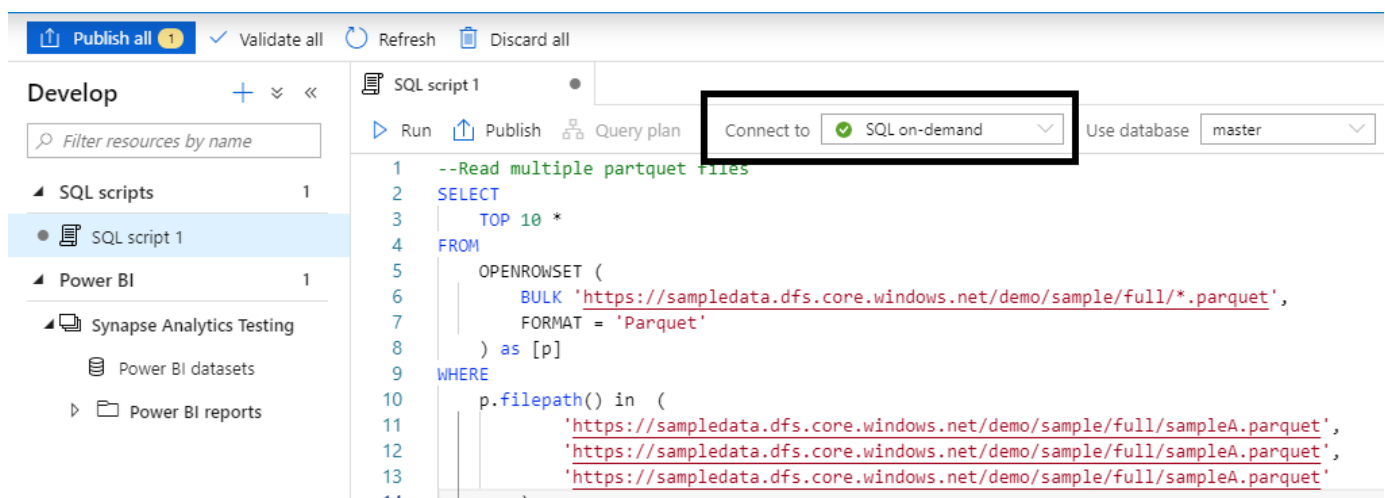


Figure 2: SQL Analytics On-Demand

Via the on-demand SQL endpoint provided in the Azure Synapse workspace, data developers can also utilize tools such as SQL Server Management Studio (SSMS) and Azure Data Studio with the on-demand compute engine.

Azure Synapse offers the flexibility to either provision and elastically scale pools of compute resources or to leverage serverless capabilities for on-demand compute resources for Azure SQL Database. With Azure Synapse, organizations can dramatically simplify the management of their data environments and bring together teams of data professionals, including data engineers, data scientists, BI professionals, and IT administrators, thus increasing collaboration and productivity.

Benefits of Azure Synapse for Power BI

Power BI professionals responsible for producing solutions that deliver actionable insights and data exploration experiences can benefit from Azure Synapse in several different ways. The following sections summarize some of the opportunities and benefits of using Azure Synapse for new and existing Power BI solutions.

Single source of truth

Building on the successful legacy of Azure SQL Data Warehouse, organizations can deploy Azure Synapse as a single, certified source of truth for Power BI and other applications. By utilizing the formally sanctioned data warehouse objects stored in provisioned SQL pools, Power BI developers and consumers of Power BI solutions can be confident that the data being presented has been validated for quality, consistency, and accuracy.

For example, Power BI administrators and other BI stakeholders may insist that only those Power BI datasets built exclusively against Azure Synapse will be eligible to be marked as a Power BI [certified dataset](#) or published to a production Premium capacity. Power BI datasets that access other, less-trusted sources, including files and legacy systems, may be limited to smaller, ad hoc scenarios.

DirectQuery at scale

Most data sources supporting DirectQuery connectivity for Power BI have historically struggled to deliver both the high user concurrency and the low query response times required for enterprise Power BI solutions. Power BI reports are designed for interactive data exploration user experiences, and this implies a high volume of queries per user session to update the different visualizations in real time. As the volume of concurrent user engagement grows into the thousands, such as with widely adopted enterprise BI solutions, common data warehouse systems such as AWS Redshift and Google BigQuery either place incoming queries into a queue, thus delaying execution, or force the user's queries to fail.

Azure Synapse supports performance optimizations, including materialized views and result set caching, to make DirectQuery models a more feasible option for vast source datasets and supporting thousands of concurrent users. With independent and elastic compute and storage resources, IT professionals can apply standard Azure resource management practices to scale provisioned SQL pools to align with the requirements of the workload. For example, simple Azure Automation runbooks could be scheduled to scale up a SQL pool to a data warehouse service level of DW3000 at 8:00 AM to support peak usage of Power BI, but then scale back down to a DW1000 level at 3:00 PM to manage costs.

Azure Synapse also offers great alternatives for Power BI model development. Assuming that recommended practices at the data source, model, and report layers are followed, Power BI professionals with access to Azure Synapse can collaborate with other data teams to deploy DirectQuery models at scale. As an example of this collaboration, data engineers could analyze the query patterns and source tables accessed by a Power BI solution and look to optimize these structures by persisting (storing and retrieving) required business logic and implementing an [ordered clustered columnstore index](#).

Organizations have naturally wanted to avoid the data movement or copying associated with the scheduled refresh and management overhead of import models. However, the need for performance at scale has driven many organizations to pursue large in-memory models to deploy to resources with sufficient RAM, such as Azure Analysis Services. For reasons of concurrency and BI performance requirements, the use of Power BI DirectQuery against Azure SQL Data Warehouse was identified as an anti-pattern by the SQL Customer Advisory Team in 2017.

Centralized security

Power BI professionals typically secure their solutions by implementing row-level security roles into data models and controlling which users or groups have access to workspaces, applications, and datasets. Azure Synapse supports both row- and column-level security for users and groups among its other layers of security features, including transparent data encryption. Although row-level security in Power BI is powerful and typically required for data models with imported data, enterprise IT organizations would generally prefer to fully leverage their data warehouse for both query processing (that is, DirectQuery) and data security.

Given that Power BI authentication is handled through Azure Active Directory (Azure AD) and given that Azure AD authentication is supported and recommended for Azure Synapse, organizations have the option to enforce data security at the data tier layer in Azure Synapse for their Power BI solutions. The identity of Power BI users and their membership in specific security groups in Azure AD can be passed to Azure Synapse so that security policies defined in Azure Synapse for the given group and source objects are enforced.

As shown below, Power BI developers can easily configure their published Synapse-based DirectQuery models to pass the credentials of the user to the data source:

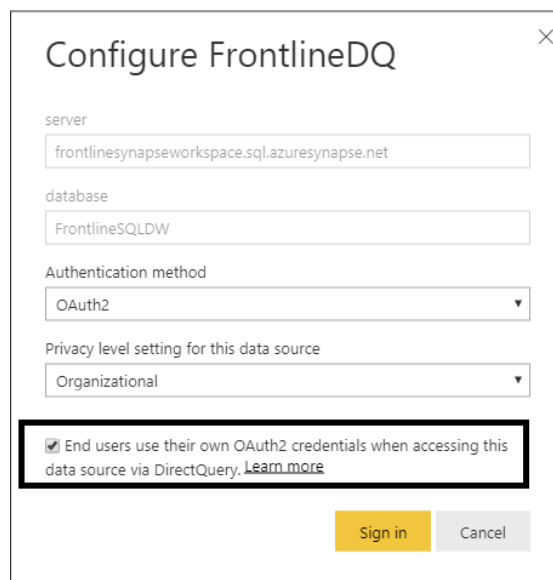


Figure 3: Single sign-on for DirectQuery connection

With data security policies handled by Azure Synapse, the risk of Power BI data models not being properly secured is eliminated in full DirectQuery mode. Additionally, since large Power BI environments typically involve many data models at varying scopes and levels of maturity, the developers and owners of these models do not have to replicate and test row-level security roles.

Composite models involving multiple storage modes (such as DirectQuery and Import) per table and (optionally) multiple data sources cannot be secured via single sign-on to a single DirectQuery data source. For example, to optimize performance for common queries, Power BI teams may choose to import an aggregated table while keeping large, detailed tables in DirectQuery mode. Additional details on composite models and aggregations are included at the end of this guide.

Team collaboration

Business intelligence has traditionally been hampered by the problems inherent with distinct teams and technologies working together toward a common goal. A team that works on data transformation processes, for example, is often unfamiliar with how these processes impact downstream applications such as Power BI. The ability to clearly communicate across teams is critical to delivering intended results in a timely manner.

Azure Synapse brings together data tools and teams, enabling greater transparency and productivity across companies. Specifically, all teams utilizing Azure Synapse access a common user interface in the Azure Synapse studio, and so all users, regardless of their primary tools or skills, are able to view and analyze the same data.

In the Azure Synapse studio, the web-based portal is accessible from an Azure Synapse workspace in Azure, multiple data development experiences are available, including Power BI reports and datasets:

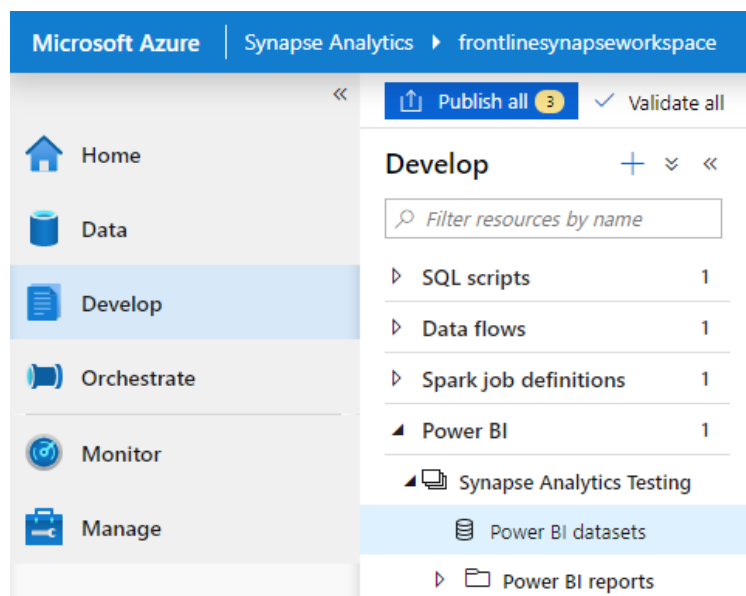


Figure 4: The Azure Synapse studio

For example, teams responsible for the data pipelines that load SQL pools would generally utilize the Orchestrate page, while data scientists, big data engineers, and Power BI developers could utilize the Data and Develop pages to access the tools and artifacts associated with their roles. With the Azure Synapse studio, teams and tools are unified in a common portal, driving more productive collaboration than ever before.

Data preparation

Power BI solutions often contain embedded data transformation and integration processes such as with Power Query, dataflows, or calculated DAX columns and tables. These transformation processes, while useful for short-term and smaller-scale scenarios, can introduce significant risks to the scalability and sustainability of the solution. The robust data processing tools of Azure Synapse, along with the expertise of Azure Synapse data engineers, can address the data preparation needs of Power BI solutions.

Azure Synapse includes the enterprise-grade data transformation and orchestration capabilities of Azure Data Factory. Data engineering teams can construct robust data pipelines, Synapse Spark jobs, or SQL stored procedures to address various data preparation needs, thereby eliminating the need for Power BI developers to handle these requirements within their solutions. The rich data processing capabilities of Azure Synapse enables Power BI developers to reallocate their efforts toward other aspects of their solutions, such as analytics, user experience, and distribution.

Paginated report flexibility

Paginated reports developed with Power BI Report Builder are an important service in Power BI environments, particularly given their strengths in exporting or printing large volumes of data. Paginated reports targeting detailed levels of data—such as individual sales orders—can be a great complement to Power BI reports and dashboards at more aggregated levels. Additionally, given access to the same SQL queries, the fine-grained controls available in Power BI Report Builder make it possible to largely replicate almost any report developed by other enterprise reporting tools.

Given full support for Azure Synapse, including basic and single sign-on authentication methods, Power BI paginated report developers have the option to build reports with common T-SQL queries directly against the provisioned SQL pool. This option is particularly valuable to expedite the migration of legacy SQL Server Reporting Services (SSRS) containing SQL queries to Power BI as well as other SQL-based reporting tools.

Building Power BI solutions with Azure Synapse

Power BI is a robust analytics platform consisting of several distinct BI artifact types, including enterprise-grade semantic models, interactive reports and dashboards, paginated reports, and self-service data transformation processes and predictive models. Azure Synapse can serve as the performant, secure, and trusted data source for each of these diverse artifacts, as well as an integrated web-based development environment.

The following sections walk through the essentials of obtaining access to an Azure Synapse resource, connecting Azure Synapse to Power BI workspaces, and developing content in either the Azure Synapse studio or utilizing Azure Synapse as a data source.

Accessing an Azure Synapse workspace

The Azure Synapse studio is the integrated web-based development and management hub for all Azure Synapse resources. All development and management activities supported by Azure Synapse are carried out in the Azure Synapse studio via access to an Azure Synapse workspace. Additionally, common development and management tools, such as SQL Server Data Tools (SSDT) for Visual Studio, SSMS, and APIs, can be used to interface with Azure Synapse resources.

Access to an Azure Synapse workspace is managed by the same [role-based access controls](#) (RBAC) applied to all other Azure resources. Therefore, to enable Power BI developers to launch the Azure Synapse studio and to access or build Power BI content from within the Azure Synapse studio, the developers need to be granted the required permissions to the Azure Synapse workspace.

Users with access to the Azure Synapse workspace will be provided with the Workspace web URL available on the Overview blade of the Azure Synapse workspace resource, as shown in *Figure 5*:

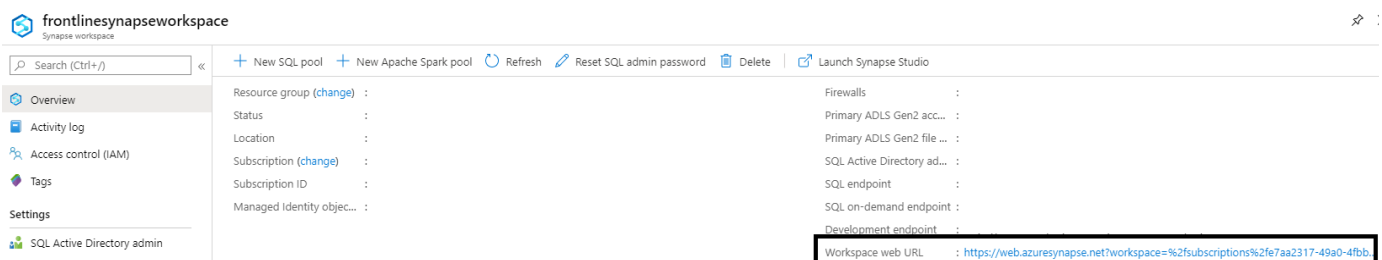


Figure 5: Workspace web URL

From the Manage blade in the Azure Synapse workspace, admins of the workspace can add users or Azure AD security groups with varying levels of permissions to the resources and artifacts in the workspace.

Administrators should be aware that mapping users or groups to a role for the workspace itself, and not the workspace Azure resource, is required for users to access the Azure Synapse studio. In *Figure 6*, both a user and a security group of users (Power BI Developers) are granted the admin roles of an Azure Synapse workspace via the Access control page for the workspace:

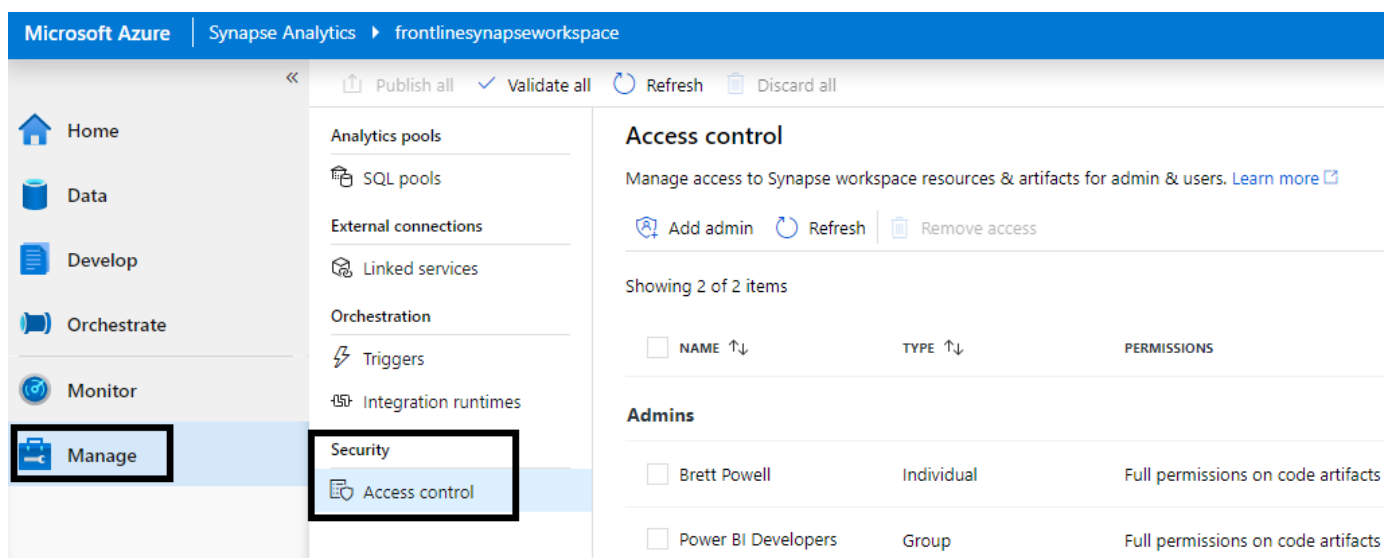


Figure 6: Workspace access control

A common and simple approach for providing user access is to map a security group of users to a built-in RBAC role, such as a contributor scoped specifically to the resource. Another common and more granular method of granting permissions is to create and manage custom role definitions that only contain the required [Azure resource operations](#). Specifically, an administration team that manages Azure resource access could identify the available operations for Azure Synapse via Azure PowerShell (`Get-AzProviderOperation`) and grant a custom role only to the operations required for Power BI development.

Workspace vs. resource access

It's important to distinguish access to the Azure Synapse workspace from access to a resource provisioned within the workspace, such as a SQL pool. Access to the Azure Synapse workspace, as described in the previous section, is only required if Power BI users will be developing Power BI content in the Azure Synapse studio or utilizing other features in the Azure Synapse studio, such as developing scripts or notebooks with SQL, Python, or other supported languages.

Typically, Power BI developers responsible for building data models, reports, and dashboards against a data warehouse are only granted read access to the source database. Most enterprise IT organizations follow strict least-privileges policies governing access to Azure resources and so, at least in the initial launch, may continue to restrict Power BI developer access to only required data sources, such as a database on a SQL pool. BI and cloud architecture teams can determine whether the benefits of the Azure Synapse studio for Power BI users described in this guide warrant providing this additional access. For example, if the Power BI developers also regularly author SQL queries and/or collaborate with data engineers, then access to the Azure Synapse studio may be particularly beneficial.

Connecting to Power BI in the Azure Synapse studio

Once access has been granted to the Azure Synapse workspace, it's necessary to establish connections from the Azure Synapse workspace to relevant Power BI app workspaces. Connections to these workspaces are defined as linked services in Azure Synapse and enable users to create and modify Power BI workspace content directly from within the Azure Synapse studio.

There are two methods available for establishing a linked service to Power BI. The most intuitive method is to click the Visualize icon from the Home pane of the workspace, as shown in *Figure 7*:

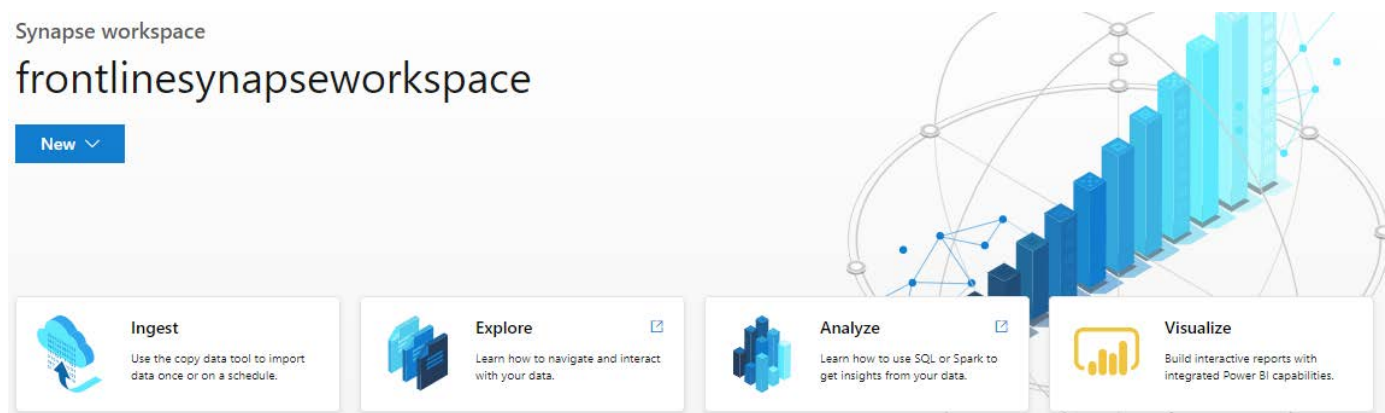


Figure 7: Synapse workspace home pane

The Visualize icon launches a form enabling the user to enter the Power BI app workspace to link to along with the name and description of the linked service. For example, in *Figure 8*, a new linked service is created with a connection to the Synapse Analytics Testing app workspace in Power BI:

Connect to Power BI

i Choose a name for your linked service. This name cannot be updated later.

Connect a Power BI workspace to create reports and datasets from data in your workspace.
[Learn more](#)

Name *
PBI Synapse Analytics Testing Workspace

Description
Power BI direct query testing over Synapse SQL pool

Workspace name *
Synapse Analytics Testing (b96f5546-1c68-4949-974e-40e1e8e8509e)

Annotations
+ New

Figure 8: Creating a linked service for Power BI

Once the linked service is created, the Azure Synapse workspace will have read and write capabilities against the Power BI app workspace. All linked services of the workspace are visible via the Linked services page, which is accessible from the Manage pane (the toolbox icon) as shown in *Figure 9*:

Microsoft Azure | Synapse Analytics | frontlinesynapseworkspace

» Publish all Validate all Refresh Discard all

Linked services

Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to external resources.

+ New

Showing 1 - 3 of 3 items

NAME ↑↓	TYPE ↑↓
frontlinesynapseworkspace-WorkspaceDefaultSqlServer	Azure Synapse Analytics (formerly SQL DW)
frontlinesynapseworkspace-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2
PBI Synapse Analytics Testing Workspace	Power BI

Figure 9: Managing the linked services

The other method for creating a linked service to Power BI is via the New icon from the Linked services page, as shown in *Figure 9*. As of the time of writing, only a single linked service to Power BI can be created from an Azure Synapse workspace. Therefore, if access to a different app workspace is required, it is currently necessary to delete the existing linked service and create a new one for the other app workspace.

Creating Power BI datasets via the Azure Synapse studio

Analytical data models defined as datasets in Power BI are central to BI solutions and overall BI architectures as they can serve as a certified and performant source for many reports, dashboards, and ad hoc analysis scenarios. In the case of Azure Synapse, Power BI developers can more easily collaborate with other data professionals on the data sources and processes impacting their models.

Once a linked service to a Power BI app workspace is in place, the Azure Synapse studio makes it easy to create a Power BI dataset file (.pbids) containing metadata for the required data source provisioned in Azure Synapse. Opening the dataset file in Power BI Desktop exposes the objects of the data source in the familiar Power Query Editor experience.

As shown in *Figure 10*, the workspace associated with the linked service is exposed on the Develop pane with the option to create a new dataset in this workspace:

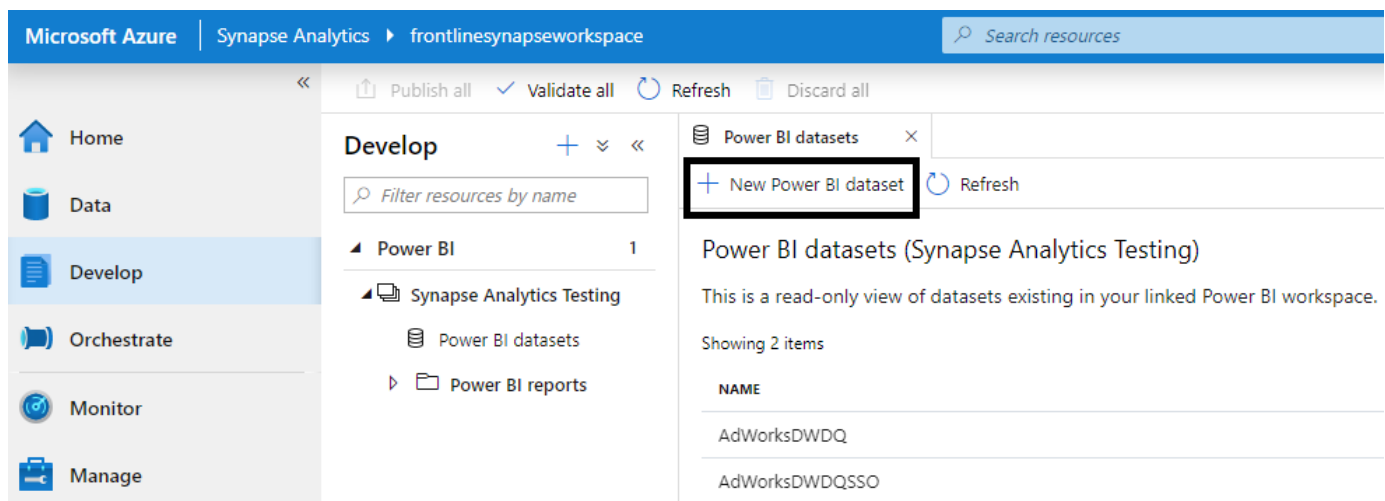


Figure 10: Creating a Power BI dataset

The New Power BI dataset form requires a data source from the workspace to be selected and, with the source selected, provides a link to download the dataset file. In *Figure 11*, the FrontlineSQLDW database hosted on a provisioned SQL pool resource is identified as the source for the new Power BI dataset:

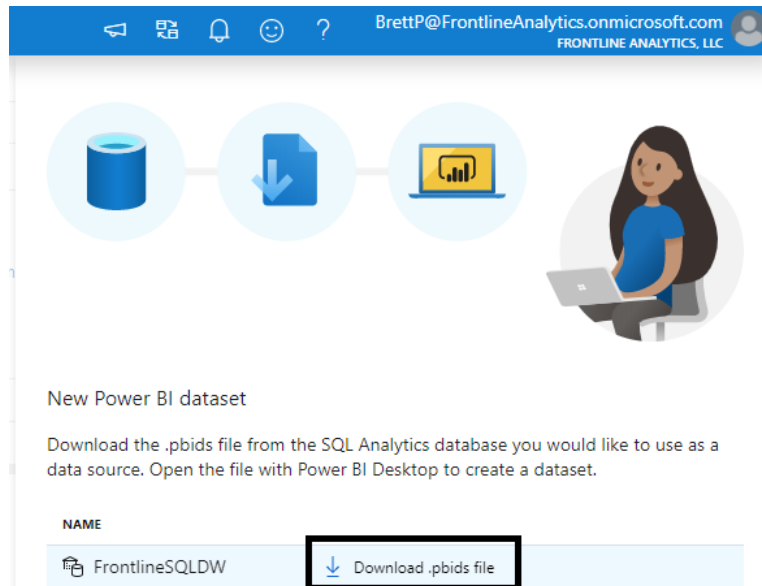


Figure 11: Downloading the dataset file

Opening the .pbids file locally with Power BI Desktop automatically launches the Navigator for the given data source, as depicted in *Figure 12*:

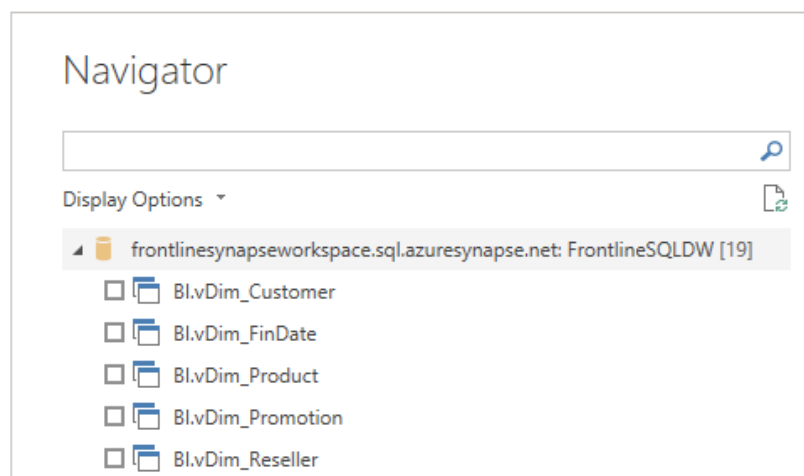


Figure 12: Opening a .pbids file in Power BI Desktop

Power BI model developers can then use common Power BI Desktop controls to modify the storage mode of the tables and further develop the relationships, metrics, and other metadata of the model. The new model can be published back to the same app workspace configured as a linked service in Azure Synapse or any other app workspace in Power BI that the user has permissions for.

As an alternative to downloading the dataset file (.pbids) from the Azure Synapse workspace, data modelers in this example could also use the Get Data experience in Power BI Desktop to define their own source connection. Specifically, the Azure SQL Data Warehouse connector found in the Azure group of data sources would be selected and the user would be required to enter the server and database names manually.

Building reports in the Azure Synapse studio

Power BI interactive reports can be created and edited directly in the Azure Synapse studio. In this example, a data model named `FrontlineDQ` has already been created and published to the Synapse Analytics Testing Power BI app workspace—the same workspace configured as a linked service in Azure Synapse. The intention is to leverage this model as the source for a new Power BI interactive report.

As shown in *Figure 13*, the plus (+) icon at the top of the Develop page in the Azure Synapse studio reveals Power BI report as an artifact that can be developed:

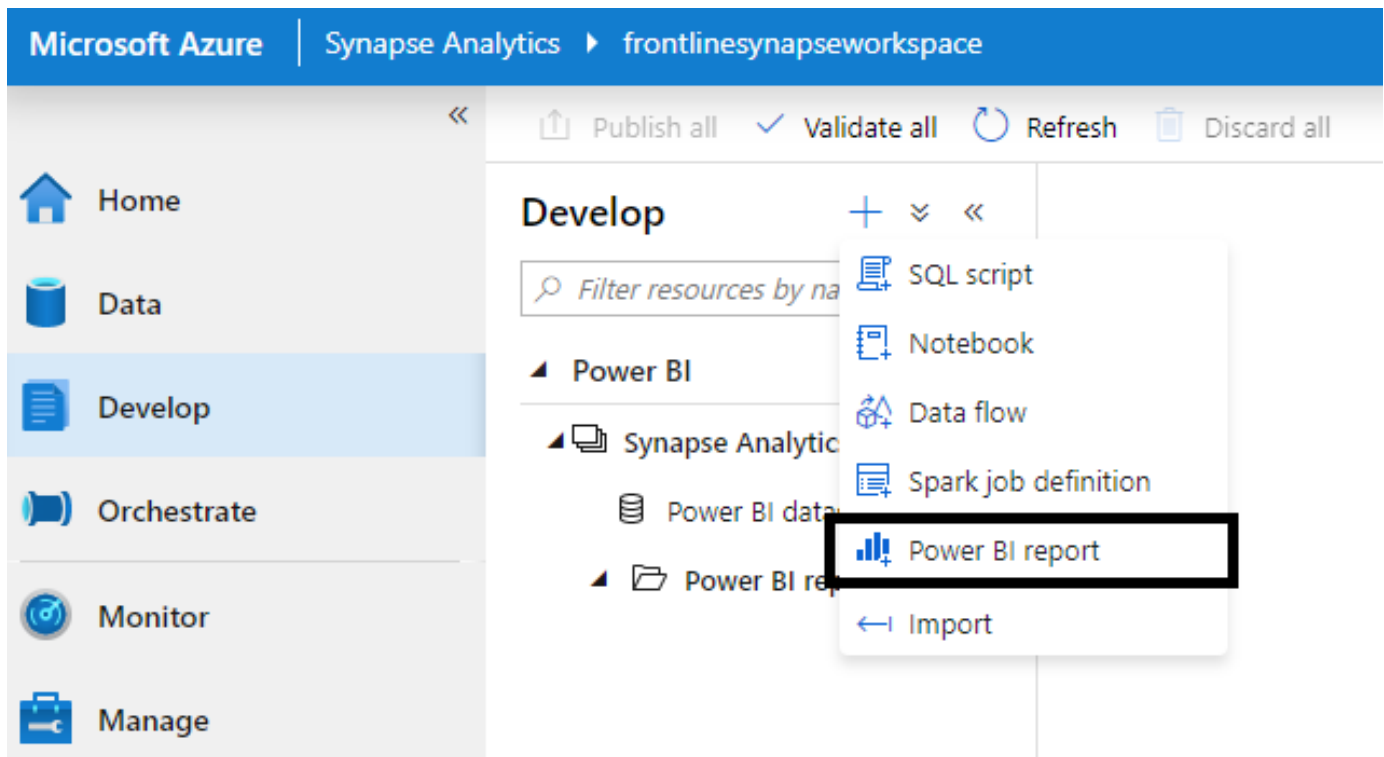


Figure 13: New Power BI report

After selecting Power BI report, the developer must then identify the Power BI dataset that will serve as the source of the new report, as shown in *Figure 14*:

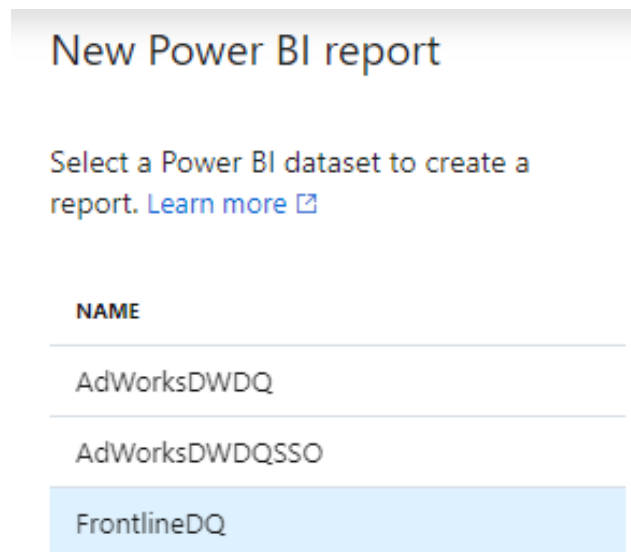


Figure 14: Available datasets for the report

Finally, clicking Create on the dataset selection form brings up the web-based Power BI report development tool from within the Azure Synapse studio. In *Figure 15*, a Power BI report named *Internet Sales* has been created within the Azure Synapse studio using the *FrontlineDQ* dataset as its source:

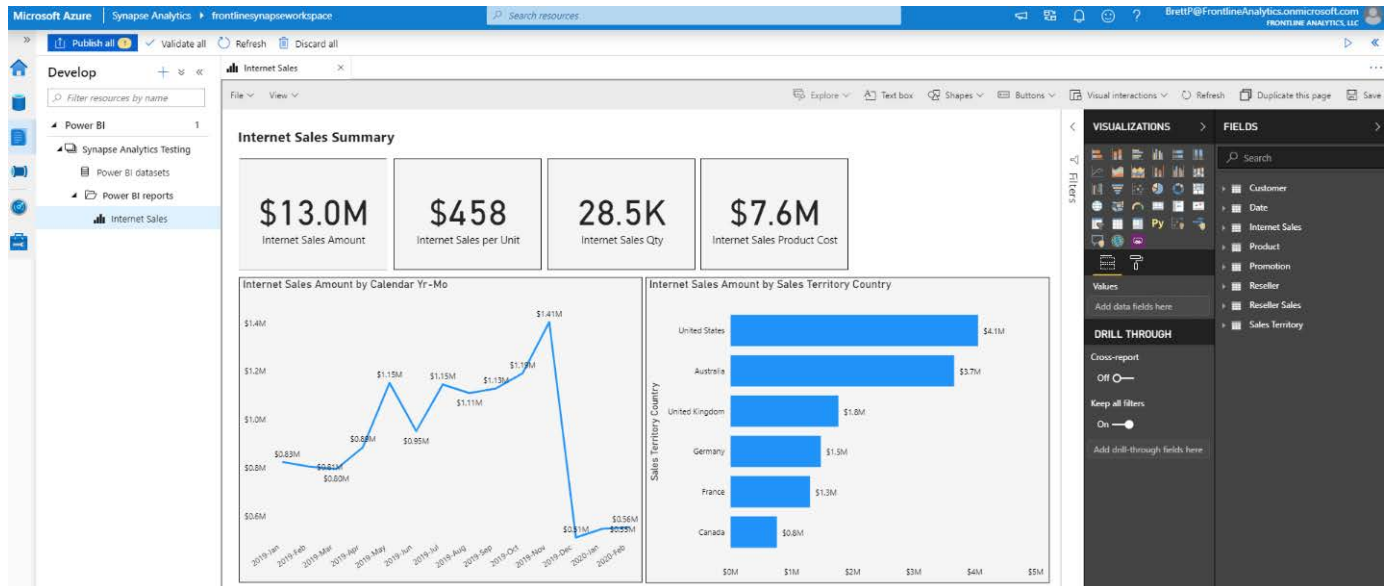


Figure 15: Creating Power BI reports in the Azure Synapse studio

As shown in Figure 15, the familiar **FIELDS**, **VISUALIZATIONS**, and **FILTERS** panes available in Power BI Desktop are shown to the report author. Likewise, Power BI reports in the Azure Synapse studio offer the same interactive experiences of filtering and cross-highlighting as Power BI and Power BI Desktop.

The completed or modified Power BI reports can be saved within the Azure Synapse studio so that they are accessible to users of the Power BI app workspace for further development or sharing with consumers of the report. (As of writing, the ability to download a copy of the report as a **.pbix** file is exclusive to Power BI.)

Although it's certainly convenient to develop Power BI reports in the Azure Synapse studio, particularly for BI developers who utilize the workspace regularly, it should be noted that this web-based development experience provides only a subset of the report development functionality of Power BI Desktop. For example, visual alignment and distribution formatting options and the ability to author report-scoped measures are currently not supported in the web experience.

As of the time of writing, given the single app workspace linked service limit of the Azure Synapse studio and the fact that the shared datasets in Power BI are not yet generally available, reports created in the Azure Synapse studio will be based on datasets in the same workspace as the linked service and saved to this same workspace.

Creating paginated reports

Unlike interactive Power BI reports, [paginated reports](#) cannot (as of this writing) be created or edited within the Azure Synapse studio. However, as described in the *[Benefits of Azure Synapse for Power BI](#)* section, paginated reports can leverage the SQL pool resource (previously Azure SQL Data Warehouse) as a secure, scalable, and performant data source. Importantly, this option enables Power BI developers to employ familiar T-SQL syntax and potentially work with existing report SQL queries and stored procedures.

Power BI dataset vs. the SQL pool

Prior to initiating the development of paginated reports, the report developer and business intelligence team should evaluate the feasibility of using an existing Power BI dataset or Analysis Services model as the source for the report. If either data model is available or can be updated to include the required data and logic of the paginated reports, then architecturally it would be preferable to reuse the relationships and calculations defined in the model for both interactive and paginated reports.

However, in some scenarios (such as highly complex or distinct report requirements), it may still be preferable or necessary to use the SQL pool resource (Azure SQL Data Warehouse) directly rather than attempt to write DAX queries against the models. Stored procedures that contain potentially complex multi-step logical processes or that expose parameters for user filtering can be created as reusable database objects in the SQL pool resource. The ability to use stored procedures and rich T-SQL syntax can be a strong factor in choosing the SQL pool as the source of reports.

Connecting to the SQL resource

To build a paginated report against the SQL pool resource, the report developer must first create a data source in Power BI Report Builder. When configuring this data source, the report developer can connect to an Azure SQL Data Warehouse data source. They'll need to use a SQL Server authentication credential as Azure AD authentication is not currently supported in Power BI Report Builder.

In *Figure 16*, a data source is defined with the Azure SQL Data Warehouse connection type and the same name as the database in the SQL pool:

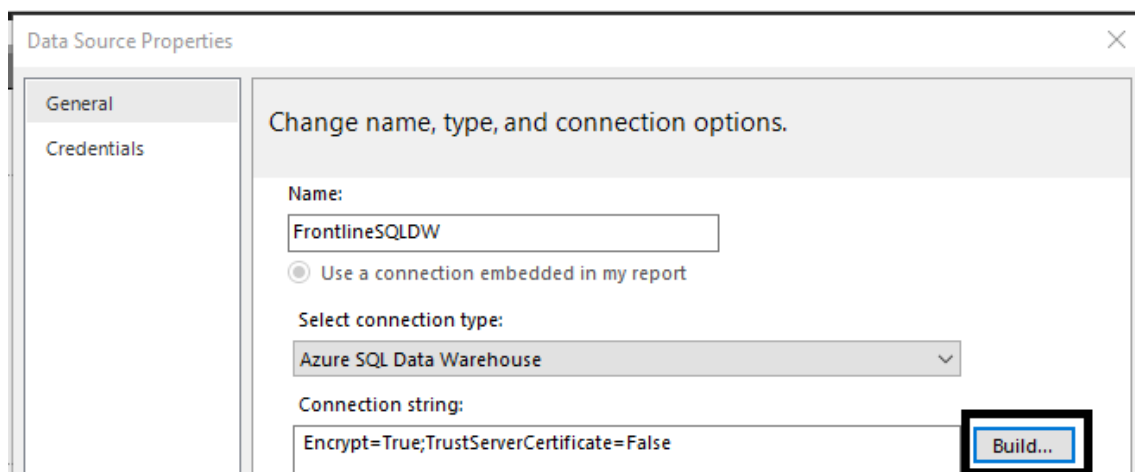


Figure 16: Paginated report data source

Clicking Build from the Data Source Properties page reveals options to provide the server name, database name, and credentials to use. This authentication information is not included in the paginated report file (.rdl) and should be securely provided by the team administering the SQL resource.

As per *Figure 17*, a SQL Server authentication credential is required to connect to the Azure SQL Data Warehouse resource from Power BI Report Builder:

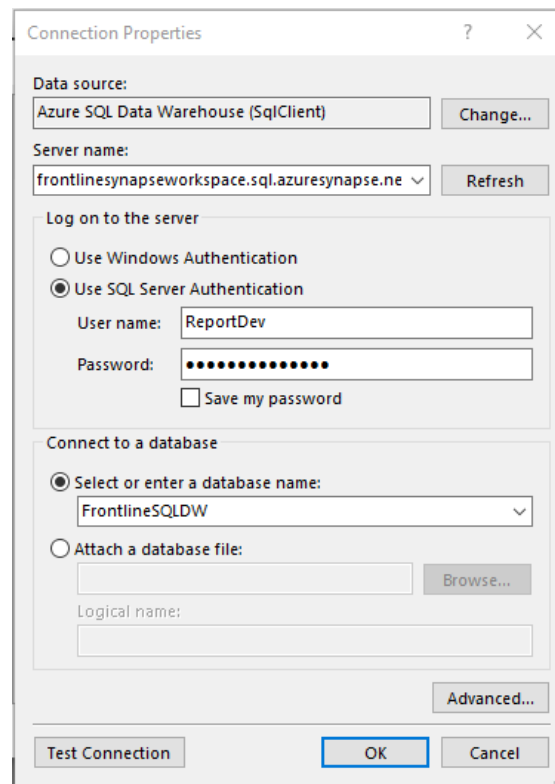


Figure 17: Data warehouse connection properties window

With a data source created and configured, the paginated report developer can define datasets for the report either via SQL statements or by referencing a stored procedure in the source database. In *Figure 18*, the `BI.spCustomersSalesOrders` stored procedure object in the `FrontlineSQLDW` resource is used as the dataset in the report:

Dataset Properties

Choose a data source and create a query.

Name:
OnlineSalesOrders

☒ Use a dataset embedded in my report.

Data source:
FrontlineSQLDW New...

Query type:
☐ Text ☐ Table ☒ Stored Procedure

Select or enter stored procedure name:
Bl.spCustomerSalesOrders fx

Query Designer... Refresh Fields

Figure 18: Using a stored procedure for a paginated report

The report author has the option to select Text and simply enter or paste in an existing SQL statement, or they can open Query Designer to use a graphical interface to help define the query. It's generally recommended to employ stored procedures in paginated reports when possible to improve the manageability of reporting solutions.

Once the paginated report is published to the Power BI service, the report author can optionally configure the authentication for the report to pass the identity of the user viewing the report.

Paginated reports built against Analysis Services models and Power BI datasets in Power BI Report Builder issue MDX queries against these sources. Although it is possible to define custom DAX queries and/or utilize the Query Designer graphical interface, Power BI Report Builder has relatively limited support for DAX-based report authoring. For example, simply configuring multi-select parameters in a report against a tabular model or Power BI dataset involves significant workarounds with custom DAX code.

Developing dataflows

Power BI dataflows are a self-service ETL capability targeting business users and are exclusively created and managed in Power BI. However, similar to paginated reports, Azure Synapse can be a common and robust data source to be utilized in dataflows to further enhance and integrate the data.

Prior to developing any dataflows against Azure Synapse, remember that self-service data preparation (and the risks to version control that these processes create) is something that Azure Synapse and data warehousing generally tries to avoid. For example, rather than a business analyst creating a dataflow to merge, cleanse, and enhance data sources, an enterprise-grade pipeline developed by a training data engineer may be a better long-term solution. All that said, resources are often too scarce to capture the requirements of new and changing data transformation scenarios or to build, test, and deploy the necessary pipelines or processing jobs. Power BI dataflows can be used to help bridge this gap to provide a less technical but nonetheless scalable self-service ETL option.

To create a dataflow against an Azure Synapse resource, navigate to an app workspace in the Power BI service. Select Dataflow from the Create drop-down menu and then select the Add new entities option as seen to the left in *Figure 19*:

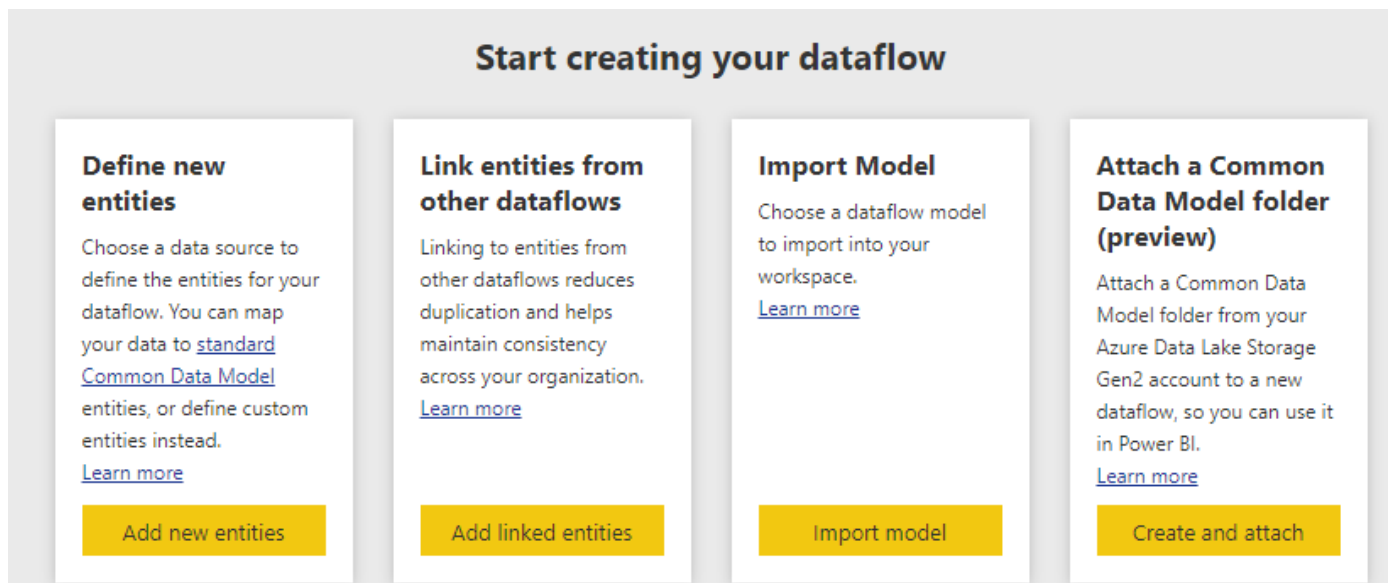


Figure 19: Creating a dataflow in Power BI

This launches the available data sources supported by Power BI dataflows. Navigate to the Azure category and select Azure SQL Data Warehouse, as shown in *Figure 20*:

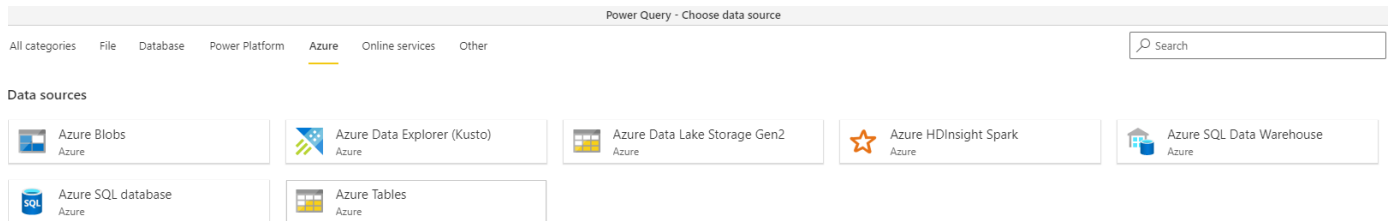


Figure 20: Azure sources for dataflows

Just like data source configuration in Power BI Report Builder, as of this writing only basic SQL Server authentication is supported by dataflows for Azure SQL Data Warehouse. In *Figure 21*, a SQL authentication login credential is used to connect to the SQL pool resource of Azure Synapse:

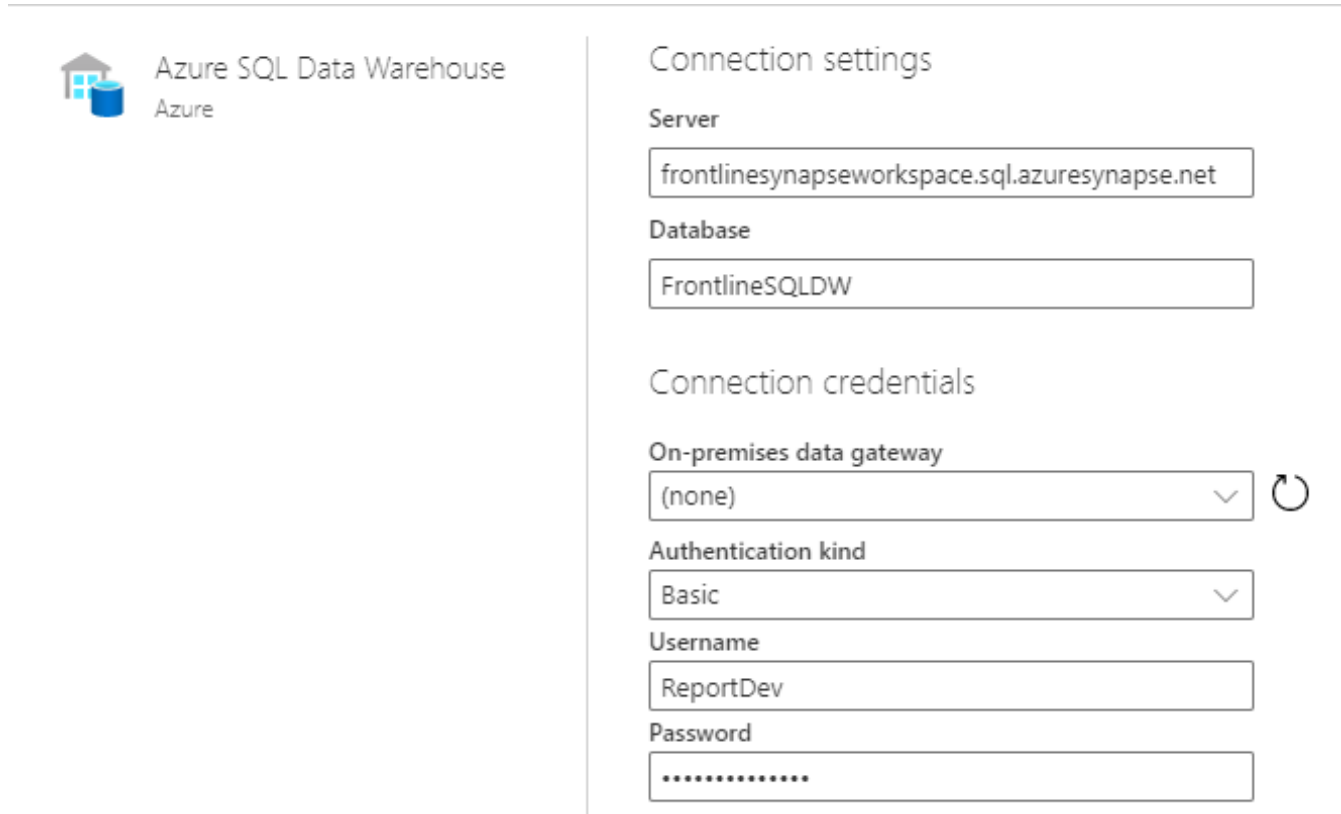


Figure 21: Azure SQL Data Warehouse dataflow authentication

Clicking Next from the Connection settings page launches the Power Query online navigation and transformation function for the SQL data source, as shown in *Figure 22*:

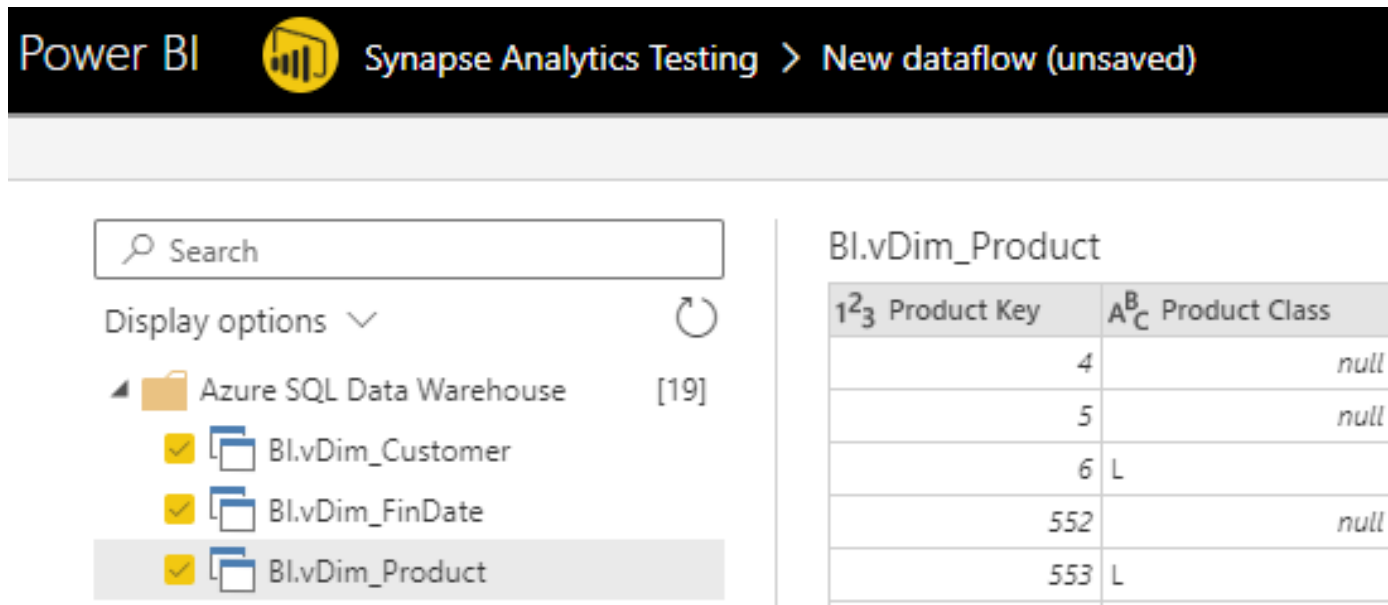


Figure 22: Power Query online navigation

Just like with the Get Data experience in Power BI Desktop, the dataflow author can select the required entities and then optionally implement any number of data transformations to enhance the value of the source data. As shown in *Figure 23*, the familiar Power Query Editor ribbon and its rich set of transformation options is available against the Azure Synapse resource:

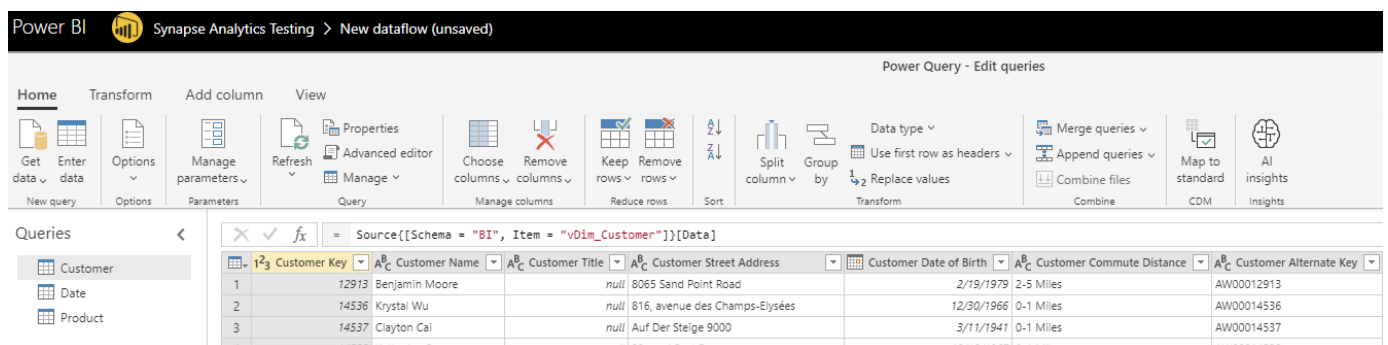


Figure 23: Dataflow entity transformations

AI predictive analytics integration

In addition to datasets, reports, paginated reports, and dataflows, Power BI also supports AI-powered predictive models and integration with Azure Machine Learning. Azure Synapse also includes Azure Machine Learning as one of its integrated services accessible in the Azure Synapse studio.

This deep integration with Azure Machine Learning, typically utilized by data scientists, along with [Automated Machine Learning in Power BI](#), enables Power BI developers and analysts with the option to either leverage the predictive models created by data scientists or to utilize the self-service model creation features available in Power BI. Specifically, Power BI analysts can build or reuse dataflows (as described in the previous section) as a means to train a predictive model. Once trained and validated, the predictive models in Power BI can be applied to other dataflows to add predictive values to incoming data.

Composite models and aggregations

Two of the most powerful data modeling features available with Power BI datasets are [composite models](#) and [aggregations](#). Whether used in isolation or in tandem, these two features give BI teams the flexibility to balance the benefits of both import and DirectQuery storage modes across multiple data sources in the same semantic model. When thoughtfully designed in collaboration with data warehouse teams, Power BI models can simultaneously deliver the query performance of compressed in-memory data caches and small, aggregated tables along with the endless scalability and data freshness of sources such as Synapse SQL pools.

Azure Synapse lends itself to common data modeling scenarios addressed by both composite models and aggregations. The following sections provide examples of these scenarios.

Targeted performance via aggregations

Business intelligence teams generally have a good understanding of the top business questions their semantic models address as well as the usage patterns and priorities of users. For example, although a fact table representing sales data may be related to seven-dimension tables, it is often the case that two or three of the dimensions are rarely utilized in reports or ad hoc analyses. Additionally, although the granularity two or three of the sales fact table may support queries at the individual customer and product levels, users may rarely drill down to this level of detail. This common mismatch between the data model and the types of queries it usually receives results in suboptimal performance and excessive resource costs to process and store the data in memory.

Aggregations enable the data modeler to embed hidden aggregated tables in their models, reflecting groupings of the most commonly used dimensions and facts of the model. With relationships defined between the aggregation table(s) and dimensions in the model, Power BI dynamically determines whether the incoming queries can be resolved by the aggregated table or whether it is necessary to query the more granular detail table.

Since an aggregated table is much smaller than a detail table, and since the aggregated table can optionally be stored in a compressed in-memory cache, the queries it resolves can achieve great performance. Moreover, in addition to aggregation tables not being visible to users, the same row-level security roles defined in models to restrict user access also apply to aggregations.

As an example, assume that the internet sales fact table in a SQL pool contains more than 5 billion rows. The amount of memory required to store this table in Power BI Premium capacity makes DirectQuery the only feasible option. However, despite performance optimizations applied to this source table, querying this table over DirectQuery connections may not deliver the desired user experience in Power BI.

To optimize the performance of common and/or highly valued queries regarding customers and sales territories over time, a new table can be created in the SQL pool database representing a grouping of the internet sales table by order date key, customer key, and sales territory key. The new aggregated table represents only a small fraction of the size of the internet sales table and can be supported as part of standard nightly data warehouse load processes and optimized for performance with the [clustered columnstore index](#), in the same way as other fact tables.

In *Figure 24*, the Manage aggregations form in Power BI Desktop allows the model author to define the summarizations and mappings to corresponding detail table columns for each column in the aggregated table:

✕

Manage aggregations

Aggregations accelerate query performance to unlock big-data sets. [Learn more](#)

Aggregation table Precedence ⓘ
InternetSalesAgg ▼ 0





AGGREGATION COLUMN	SUMMARIZATION	DETAIL TABLE	DETAIL COLUMN	
CustomerKey	GroupBy ▼	Internet Sales ▼	CustomerKey ▼	
OrderDateKey	GroupBy ▼	Internet Sales ▼	OrderDateKey ▼	
Sales_Summed	Sum ▼	Internet Sales ▼	SalesAmount ▼	
SalesTerritoryKey	GroupBy ▼	Internet Sales ▼	SalesTerritoryKey ▼	

Figure 24: Manage aggregations

By default, once an aggregation is applied to the table, its metadata is hidden from the user interface.

The model author then simply relates the three columns defining the granularity of the aggregate table to their corresponding dimension columns, resulting in the schema depicted in *Figure 25*:

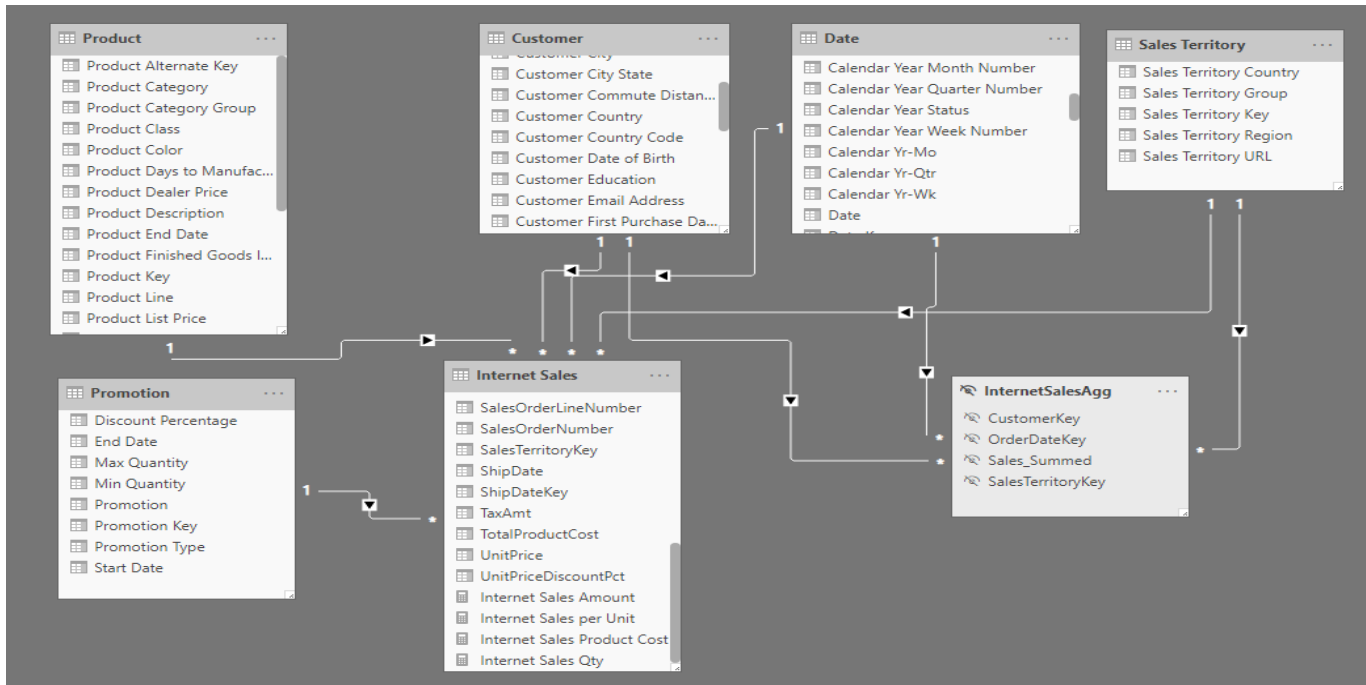


Figure 25: Model with aggregations

In the updated Power BI dataset, all BI queries from Power BI and other tools that require the sum of the sales amount column by the Customer, Date, or Sales Territory dimensions, or some combination of these three-dimension tables, are resolved by the smaller *InternetSalesAgg* table and therefore result in enhanced performance. Queries that request columns from the *Promotion* or *Product* dimension tables, however, still utilize the *Internet Sales* table.

Table storage mode

A primary feature of composite models and aggregations is the ability to define the storage mode for each table in a Power BI model. Via the Storage mode property in the Advanced card in the Modeling view of Power BI Desktop, as shown in *Figure 26*, model designers can choose between Import, DirectQuery, and Dual:

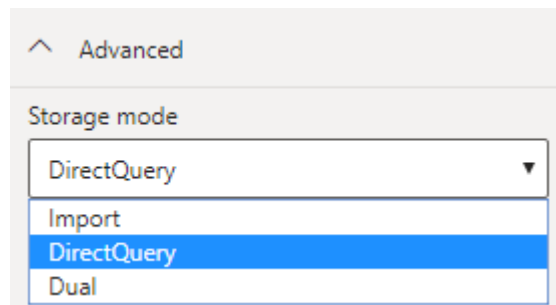


Figure 26: Storage mode settings

In the case of the `InternetSalesAgg` aggregations table from the earlier example, the business intelligence and data warehouse team could decide whether this table should also be DirectQuery, like all other tables in the model, or whether it should be imported into a compressed memory cache in Power BI. This decision has significant implications for both performance and security.

From a performance perspective, an imported table delivers varying degrees of improved query performance over the same table in DirectQuery mode due to the optimizations built into Power BI's in-memory columnar engine. However, the aggregated table in DirectQuery mode can still represent a dramatic performance improvement relative to queries against the multi-billion row fact table.

In terms of security, keeping all tables in DirectQuery mode enables the team to leverage the security built into the source SQL pool and single sign-on authentication, which passes the user's identity to the source. With import and dual storage mode tables defined in the model, row-level security roles would need to be defined and managed in the Power BI model. As always, business intelligence teams have to weigh these factors relative to their environments and requirements.

Blending sources and connectivity

Similar to how aggregations dynamically determine which tables to query, Power BI models can also query multiple data sources, potentially with different storage modes. Power BI is responsible for querying both sources and combining the results to support the required visualizations. For example, at a certain stage in a project, a specific table required for a model may only be available as a .csv file. Via composite models and table storage modes, this .csv file could be added to a model that already contains several tables with DirectQuery connections to a SQL pool. Additionally, model relationships can be created between the .csv-based table with the DirectQuery-based SQL pool tables to control the cross-filtering behavior in report visualizations. Composite models allow teams the flexibility to utilize multiple data sources and alternative storage modes.

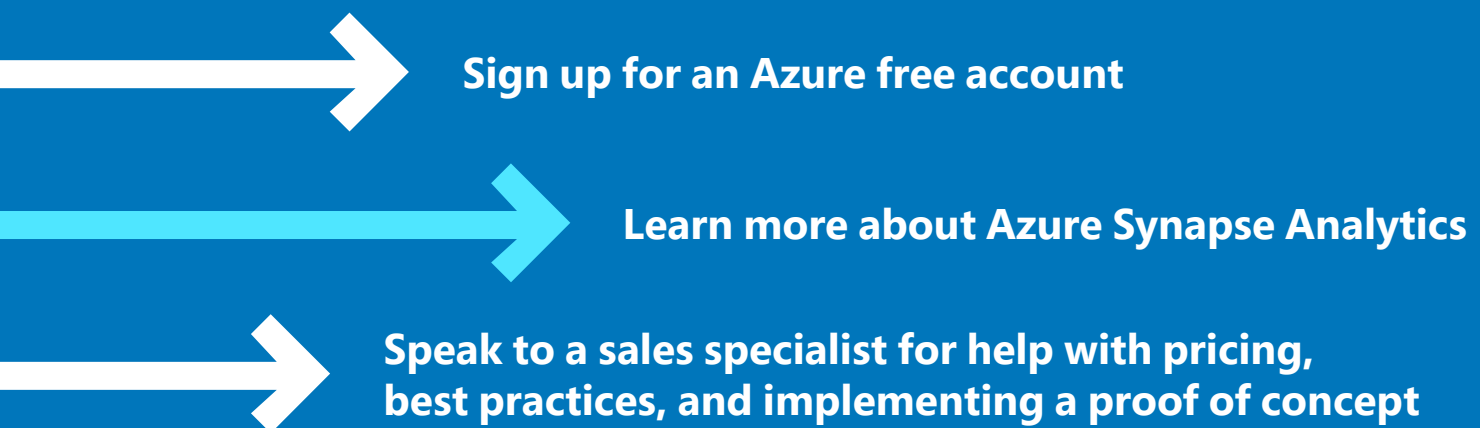
Conclusion

Azure Synapse Analytics and Power BI become more powerful when used together, combining to provide a unique, modern approach to data analytics. Azure Synapse empowers Power BI professionals across a diverse set of use cases to deliver the scale, performance, and cost management your projects require. Interactive Power BI reports and enterprise-grade semantic models can be developed within the Azure Synapse studio, the new common web portal for developing and managing various Azure Synapse artifacts.

Some of the key benefits of Azure Synapse for Power BI professionals are that it:

- Acts as a single, certified source of truth for Power BI
- Supports performance optimizations enabling DirectQuery at scale
- Supports row- and column-level security along with other integrated security features
- Facilitates team collaboration and transparency through a common user interface
- Includes enterprise-grade data transformation and orchestration capabilities for robust data preparation
- Provides flexible support for building paginated reports with Power BI Report Builder

Sign up for an Azure free account today to see how combining Azure Synapse Analytics with Power BI can benefit your company.



About the authors

Jack Lee is a senior Azure certified consultant and an Azure practice lead with a passion for software development, cloud, and DevOps innovations. He is an active Microsoft tech community contributor and has presented at various user groups and conferences, including the Global Azure Bootcamp at Microsoft Canada. Jack is an experienced mentor and judge at hackathons and is also the president of a user group that focuses on Azure, DevOps, and software development. He has been recognized as a Microsoft MVP for his contributions to the tech community. You can follow Jack on Twitter at [@jlee_consulting](#).

Brett Powell is the owner of Frontline Analytics, a data and analytics consulting firm and Microsoft Power BI partner. He has worked with Power BI technologies since they were first introduced with the Power Pivot add-in for Excel 2010 and has contributed to the design and delivery of Microsoft BI solutions across retail, manufacturing, finance, and professional services. He is also the author of *Mastering Microsoft Power BI* and *Microsoft Power BI Cookbook* and is a regular speaker at Microsoft technology events, such as the Power Platform World Tour and the Data & BI Summit. He regularly shares technical tips and examples on his blog, [Insight Quest](#).

