Learn Azure Synapse Data Explorer

A guide to building real-time analytics solutions to unlock log and telemetry data

Pericles (Peri) Rocha

BIRMINGHAM—MUMBAI

Learn Azure Synapse Data Explorer

Publishing Product Managers: Birjees Patel and Arindam Majumder

Content Development Editor: Shreya Moharir

Technical Editor: Devanshi Ayare

Copy Editor: Safis Editing

Project Coordinator: Farheen Fathima

Proofreader: Safis Editing

Indexer: Tejal Daruwale Soni

Production Designer: Shankar Kalbhor

Marketing Coordinator: Nivedita Singh

First published: February 2023

Production reference: 1200123

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

ISBN 978-1-80323-395-6

www.packtpub.com

To my daughter, Isabella, I love you to the moon and all the way back. To my wife, Cecilia, my partner, and the love of my life, thank you for your patience, love, friendship, and partnership in life. I love you. To my brother, Plinio, my best friend, and my favorite companion in the things we do together. And last but not least, in loving memory of my mother, Yara, and my father, Jose. This work is dedicated to all of you.

Contributors

About the author

Pericles (Peri) Rocha is a technical product manager, architect, and data scientist with more than 25 years of experience. He has worked with diverse challenges from building highly available database environments to data science projects. He holds an MSc degree in data science from UIUC and is a member of Tau Beta Pi. He currently works at Microsoft as a product manager in the Azure Synapse engineering team. Originally from São Paulo, Brazil, Peri worked in Europe for three years before relocating to the USA in 2016. In his spare time, he enjoys playing music, studying karate, and reading. He lives near Redmond, WA, with his wife, daughter, two dogs, and nine guitars.

I'd like to thank everyone who crossed my path through 25 years of professional experience. All of you helped me shape my own story and I am deeply thankful for it.

About the reviewer

Felipe Andrade is a client technical lead at Microsoft Canada. He has been at Microsoft for 9 years and has been working with data analytics for over 10 years. He has spent most of his career at Microsoft in analytics technical roles working with Power BI, SQL, Synapse, Databricks, and machine learning. He also worked in a couple of startups as a software engineer running social network analytics.

I'd like to thank Peri Rocha for inviting me to be a technical reviewer for his book. Thanks to my family, Leticia, Luisa, and Alice, for their patience and kindness.

Table of Contents

2

Creating Your First Data Explorer Pool 35

3

Exploring Azure Synapse Studio 61

4

Real-World Usage Scenarios 85

Part 2: Working with Data

5

Ingesting Data into Data Explorer Pools 101

6

Data Analysis and Exploration with KQL and Python 137

7

Data Visualization with Power BI 177

8

Building Machine Learning Experiments 195

9

Exporting Data from Data Explorer Pools 219

Part 3: Managing Azure Synapse Data Explorer

10

11

12

13

Preface

Large volumes of data are generated daily from applications, websites, internet of things devices, and other free-text, semi-structured data sources. Azure Synapse Data Explorer helps you collect, store, and analyze such data, and enables you to work with other analytical engines, such as Apache Spark, to develop advanced data science projects and maximize the value you get from your log and telemetry data.

This book offers a comprehensive view of Azure Synapse Data Explorer, covering not only the core scenarios of Data Explorer but also how it integrates into the whole picture within Azure Synapse. From data ingestion, through data visualization and advanced analytics, you will learn an end-to-end approach to maximizing the value of unstructured data and driving powerful insights using data science capabilities. With real-world usage scenarios, you'll learn how to identify key projects where Azure Synapse Data Explorer can help you achieve your business goals. You will also learn how to manage big data as part of a platform as a service offering, tune, secure, and serve data at scale to end users.

By the end of this book, you will have mastered the big data life cycle and be able to implement advanced analytical scenarios from raw telemetry and log data.

Who this book is for

If you are a data engineer, data analyst, or business analyst working with unstructured data and want to learn how to maximize the value of such data, this book is for you. To maximize your learning experience from this book, you should be familiar with working with data and performing simple queries using SQL or KQL. Even though it is not a requirement, familiarity with Python will help you get more from the examples. This book is also excellent for professionals already working with Azure Synapse who want to incorporate unstructured data into their data science projects.

What this book covers

Chapter 1, *Introducing Azure Synapse Data Explorer*, is the first of four chapters in *Part 1*, *Introduction to Azure Synapse Data Explorer*, where you will be introduced to the product and learn the basics that you need before you start to work with data. It welcomes you to Azure Synapse Data Explorer and elaborates on the need for a fast and highly scalable data exploration service for telemetry and log data. It introduces Azure Synapse and explains how the Data Explorer service fits under the Azure Synapse umbrella. Finally, it discusses the architecture and infrastructure of Data Explorer pools, and the scale of the service today.

Chapter 2, *Creating Your First Data Explorer Pool*, gets your hands busy by walking you through the creation of your first Azure Synapse workspace and a Data Explorer pool using the Azure portal, Azure Synapse Studio, or the Azure **Command-Line Interface** (**CLI**). If you are not familiar with Azure yet, don't worry; this chapter guides you through the steps to create your first free Azure account, allowing you to follow the examples in the book.

Chapter 3, *Exploring Azure Synapse Studio*, introduces the development and management environment of Azure Synapse. You will learn about the user interface elements of Azure Synapse Studio, and where to find what you are looking for by navigating through the hubs. In addition to that, in this chapter, you will load some data into a database and run your first query to help you familiarize yourself with the query editor. This chapter closes with an overview of where to manage and monitor your environment using Azure Synapse Studio.

Chapter 4, *Real-World Usage Scenarios*, describes some example solution architectures you can use in common log and telemetry data analytics scenarios. It looks at five real-world use cases that integrate Azure Synapse Data Explorer with other Azure services and helps you understand the blueprints so that you can build your own.

Chapter 5, *Ingesting Data into Data Explorer Pools*, kicks off *Part 2*, *Working with Data*. It walks you through the data loading process, choosing your own data loading strategy, and walks you through different ways to load data into Data Explorer pools. This chapter builds the data assets that you will use in most chapters of the book.

Chapter 6, *Data Exploration and Analysis with KQL and Python*, is all about learning how to query, transform, and get insights from your data using **Kusto Query Language** (**KQL**) and Python. You will learn how to use KQL to explore the data you have at hand and familiarize yourself with the schema, plot simple charts in the query editor, obtain percentiles, and even use native KQL commands to look at trends in your data using linear regression. In the second half of this chapter, you will create an Azure Synapse notebook to explore and transform data using Python and create a lake database.

Chapter 7, *Data Visualization with Power BI*, complements the previous chapter by helping you configure Power BI integration with Azure Synapse and author new Power BI reports directly from Azure Synapse Studio. It walks you through the creation of reports that connect to data in Data Explorer pools, as well as to your new lake database.

Chapter 8, *Building Machine Learning Experiments*, provides an overview of applied machine learning, and how to introduce advanced analytics to your Azure Synapse projects using **automated machine learning** (**AutoML**). You will use Python to prepare your data for machine learning experiments, train a series of models, and find the best model to help you predict values.

Chapter 9, *Exporting Data from Data Explorer Pools*, closes *Part 2*, *Working with Data*, by walking you through data export scenarios. It explains scenarios where data exports are needed and walks you through different options you have available to perform data exports, including continuous data exports.

Chapter 10, *System Monitoring and Diagnostics*, is the first of four chapters in *Part 3*, *Managing Azure Synapse Data Explorer*. In this chapter, you will learn about managing a platform-as-a-service service such as Azure Synapse, and which parts of the service you should be concerned with. Through code examples and guidance through the user interface, you will learn how to stay on top of your Data Explorer pools and proactively monitor them. By setting up alerts, you'll learn how to get notified on your phone if an event of interest happens in your environment.

Chapter 11, *Tuning and Resource Management*, introduces resources to help you provide predictable performance to end users and using cache policies to speed up queries. It walks you through the implementation of resource management to help you categorize user requests to prioritize the execution of critical workloads while queueing requests that can wait.

Chapter 12, *Securing Your Environment*, provides you with the information you need to make sure your data is secure at rest and in transit, and that only people who are intended to access your data have access to it. It walks you through an overview of the security issues you need to consider for your own implementations, how to double-encrypt your data for an added layer of security, how to authenticate and authorize users, and how to protect the network environment that transits your data.

Chapter 13, *Advanced Data Management*, covers how to adhere to governmental regulations for data handling, including how to permanently purge personal data. You will learn how to use extents, or data shards, in Azure Synapse Data Explorer to move large volumes of data quickly for archival.

To get the most out of this book

To maximize your learning experience, you should have a basic understanding of concepts around data integration, data retrieval, and building basic data visualizations. Previous experience with SQL, KQL, and Python is not required, but it will help you understand the concepts in the code examples more quickly.

Software/hardware covered in the book	Operating system requirements
Azure Synapse Studio	Windows, macOS, or Linux
The Azure portal	Windows, macOS, or Linux
Power BI Desktop	Windows
Microsoft Azure App	iOS or Android

The Azure portal and Azure Synapse Studio are web-based tools that are used to manage, develop, and build solutions for Azure Synapse Data Explorer. Microsoft supports the latest versions of the following browsers: Microsoft Edge, Safari (Mac only), Chrome, and Firefox.

To install Power BI Desktop, visit `https://learn.microsoft.com/power-bi/fundamentals/desktop-get-the-desktop`.

To install the Microsoft Azure App, visit `http://aka.ms/getazureapp` on your mobile device, or look for the Microsoft Azure App in your device's app store.

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

Download the example code files

You can download the example code files for this book from GitHub at `https://github.com/PacktPublishing/Learn-Azure-Synapse-Data-Explorer`. If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: `https://packt.link/DQQ7A`.

Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "To create or alter a new workload group, use the `.create-or-alter workload_group` command."

A block of code is set as follows:

```
.alter-merge workload_group ['Engineering Department WG'] ```
{
  "RequestQueuingPolicy": {
      "IsEnabled": true
  }
} ```
```

Any command-line input or output is written as follows:

```
az synapse kusto pool create --name "droneanalyticsadx"
--resource-group "rg-AzureSynapse" --sku name="Compute
optimized" size="Small" --workspace-name "drone-analytics"
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: "To enable it, you must select the **Enable** option next to **Double encryption using a customer-managed key**, in the **Security** tab of the **Create Synapse workspace** wizard."

> **Tips or important notes**
> Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at `customercare@packtpub.com` and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata` and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Share Your Thoughts

Once you've read *Learn Azure Synapse Data Explorer*, we'd love to hear your thoughts! Scan the QR code below to go straight to the Amazon review page for this book and share your feedback.

https://packt.link/r/1-803-23395-8

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere? Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below

`https://packt.link/free-ebook/9781803233956`

2. Submit your proof of purchase
3. That's it! We'll send your free PDF and other benefits to your email directly

Part 1
Introduction to Azure Synapse Data Explorer

To maximize your learning experience, you should quickly become familiar with the core concepts and tools you will work with when reproducing the examples and learning new concepts, and how these concepts can help you in real-life projects. The first part of the book focuses on introducing Azure Synapse Data Explorer and all of its layers. You will learn about the service architecture, all of the platform elements within Azure Synapse, and how to create your own lab environment to run through the book examples. You will also become familiar with Azure Synapse Studio, and the development and management interface of Azure Synapse. Finally, you will learn about solution templates from real-world usage scenarios that will help you speed up your own Azure Synapse Data Explorer implementations.

This part comprises the following chapters:

- *Chapter 1*, *Introducing Azure Synapse Data Explorer*
- *Chapter 2*, *Creating Your First Data Explorer Pool*
- *Chapter 3*, *Exploring Azure Synapse Studio*
- *Chapter 4*, *Real-World Usage Scenarios*

1

Introducing Azure Synapse Data Explorer

Every day, applications and devices connected to the internet generate massive amounts of data. To give some perspective, we expect to have 50 billion connected devices by 2030 generating data, and up to 175 **zettabytes** (**ZB**) of data generated by 2025 (from every possible source). As more and more new connected devices reach the market every year, and as companies make greater use of unstructured data from application logs, the amount of data generated daily will become difficult to measure. In fact, some companies are keeping certain types of data, such as telemetry and application logs, for no longer than a certain period (such as 90 to 120 days) because even with the fact that storage has never been cheaper, storing and managing large volumes of data can quickly become cost-prohibitive.

Being able to store, manage, and quickly analyze unstructured data has become a critical business need for most companies. From application logs, you can predict the behavior of users and respond quickly to user demand. By analyzing device telemetry, you can anticipate hardware failures, reduce downtime in factories, predict the weather, and detect patterns that help optimize your operation. Most importantly, the ability to correlate application and device data, apply **machine learning** (**ML**) algorithms, and visualize data in real time allows you to respond quickly to operational challenges, as well as customer and market demands.

Azure Synapse Data Explorer complements the **Synapse Structured Query Language** (**Synapse SQL**) engine and Apache Spark engine already present in Azure Synapse to offer a big data service that helps acquire, store, and manage big data to unlock insights from device telemetry and application logs. It works just like the **Azure Data Explorer** standalone service, but with the benefit of tightly integrating with the other services offered by Azure Synapse, allowing you to build **end-to-end** (**E2E**) advanced analytics projects from data ingestion to rich visualizations using Power BI.

By the end of this chapter, you should have a thorough understanding of where Azure Synapse Data Explorer fits in the data lifecycle, how to describe the service and differentiate it from the standalone service, and when to use Data Explorer pools in Azure Synapse.

In this chapter, we will go through the following topics:

- Understanding the lifecycle of data
- Introducing the Team Data Science Process
- The need for a fast and highly scalable data exploration service
- What is Azure Synapse?
- What is Azure Synapse Data Explorer?
- Integrating Data Explorer pools with other Azure Synapse services
- Exploring the Data Explorer pool infrastructure and scalability
- What makes Azure Synapse Data Explorer unique?
- When to use Azure Synapse Data Explorer

Technical requirements

To build your own environment and experiment with the tools shown in this chapter (and throughout the book), you will need an Azure account and a subscription. If you don't have an Azure account, you can create one for free at `https://azure.microsoft.com/free/`. Microsoft offers $200 in Azure credit for 30 days, as well as some popular services for free for 1 year. Azure Synapse is not one of the free services, but you should be able to use your free credit to run most examples in this book as long as you adhere to the following practices:

- **Using the smallest pool sizes**: Azure Synapse Data Explorer offers pool sizes ranging from extra small (2 cores per instance) to large (16 cores per instance). Picking the smallest pool size options will help you save money and still learn about Azure Synapse Data Explorer without any constraints.
- **Keeping your scale to a minimum**: As with pool sizes, you don't need several instances running on your cluster to learn about Azure Synapse Data Explorer. Avoid using autoscale (discussed in *Chapter 2*), and keep your instance count to a minimum of two.
- **Manage your storage**: Azure Synapse Data Explorer also charges you by storage usage, so if you're trying to save costs in your learning journey, make sure you only have the data you need for your testing.
- **Stop your pools when not in use**: You are charged for the time your cluster is running, even if you are not using it. Make sure you stop your Data Explorer pools when you are done with your experiments so that you are not charged. You can resume your pools next time you need them!

One or more examples in this chapter make use of the *New York Yellow Taxi* open dataset available at `https://docs.microsoft.com/en-us/azure/open-datasets/dataset-taxi-yellow?tabs=azureml-opendatasets`.

> **Note**
>
> The Azure free account offer may not be available in your country. Please check the conditions before you apply.

Understanding the lifecycle of data

The typical data lifecycle in the world of analytics begins with data generation and ends with data analysis, or visualization through reports or dashboards. In between these steps, data gets ingested into an analytical store. Data may or may not be transformed in this process, depending on how the data will be used. In some cases, data can be updated after it has been loaded into an analytical store, even though this is not optimal. Appending new data is quite common.

Big data is normally defined as very large datasets (volume) that can be structured, semi-structured, or unstructured, without necessarily having a pre-defined format (variety), and data that changes or is produced fast (velocity). *Volume*, *variety*, and *velocity* are known as *the three Vs* of big data.

> **Note**
>
> While most literature defines the three Vs of big data as volume, variety, and velocity, you may also see literature that defines them as five Vs: the previously mentioned volume, variety, velocity, but also veracity (consistency, or lack of) and value (how useful the data is). It is important to understand that a big data solution needs to accommodate loading large volumes of data at low latency, regardless of the structure of the data.

For data warehousing and analytics scenarios in general, you will typically go through the following workflow:

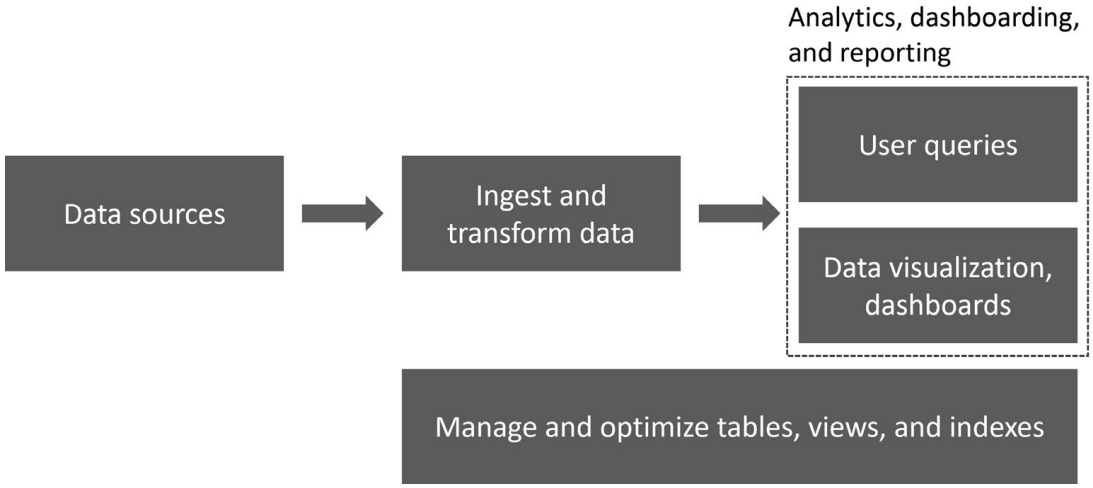

Figure 1.1 – A typical workflow in analytics

Let us break down the steps in this process, as follows:

- **Data sources**: This is where data originates from. Some examples of data sources may include a sales application that stores transactions on a database (in which case, the database in question would be the source), telemetry data from **internet of things** (**IoT**) devices, application log data, and much more.
- **Create database objects**: The first step is to create the database itself, and any objects you will need to start loading data. Creating tables at this stage is common, but not required—in many cases, you will create destination tables as part of the data ingestion phase.
- **Ingest and transform data**: The second step is to bring data to your analytical store. This step involves acquiring data, copying it to your destination storage location, transforming data as needed, and loading it to a final table (not necessarily in this order—sometimes, you will load data and transform it in the destination location) that will be retrieved by user queries and dashboards. This can be a complex process that may involve moving data from a source location to a data lake (a data repository where data is stored and analyzed in its raw form), creating intermediary tables to transform data (sort, enrich, clean data), creating indexes and views, and other steps.
- **User queries, data visualization, and dashboards**: In this step, data is ready to be served to end users. But this does not mean you are done—you need to make sure queries are executed at the expected performance level, and dashboards can refresh data without user interaction while reducing overall system overhead (we do not want a dashboard refreshing several times per day if that's not needed).
- **Manage and optimize tables, views, and indexes**: Once the system is in production and serving end users, you will start to find system bottlenecks and opportunities to optimize your analytical environment. This will involve creating new indexes (and maintaining the ones you have created before!), views, and materialized views, and tuning your servers.

The lifecycle of big data can be similar to that of a normal data warehouse (a robust database system used for reporting and analytics), but it can also be very specific. For the purpose of this book, we'll look at big data from the eyes of a data scientist, or someone who will deliver advanced analytics scenarios from big data. Building a pipeline and the processes to ensure data travels quickly from when it is produced to unlock insights without compromising quality or productivity is a challenge for companies of all sizes.

The lifecycle of data described here is widely implemented and well proven as a pattern. With the growth of the data science profession, we have observed a proliferation of new tools and requirements for projects that went well beyond this pattern. With that, came the need for a methodology that helps govern ML projects from gathering requirements up to model deployment, and everything in between, allowing data scientists to focus on the outcomes of their projects as opposed to building a new approach for every new project. Let's look at how the TDSP helps achieve that.

Introducing the Team Data Science Process

In 2016, Microsoft introduced the **Team Data Science Process** (**TDSP**) as an agile, iterative methodology to build data science solutions at scale efficiently. It includes best practices, role definitions, guidelines for collaborative development, and project planning to help data scientists and analysts build E2E data science projects without having to worry about building their own operational model.

Figure 1.2 illustrates the stages in this process:

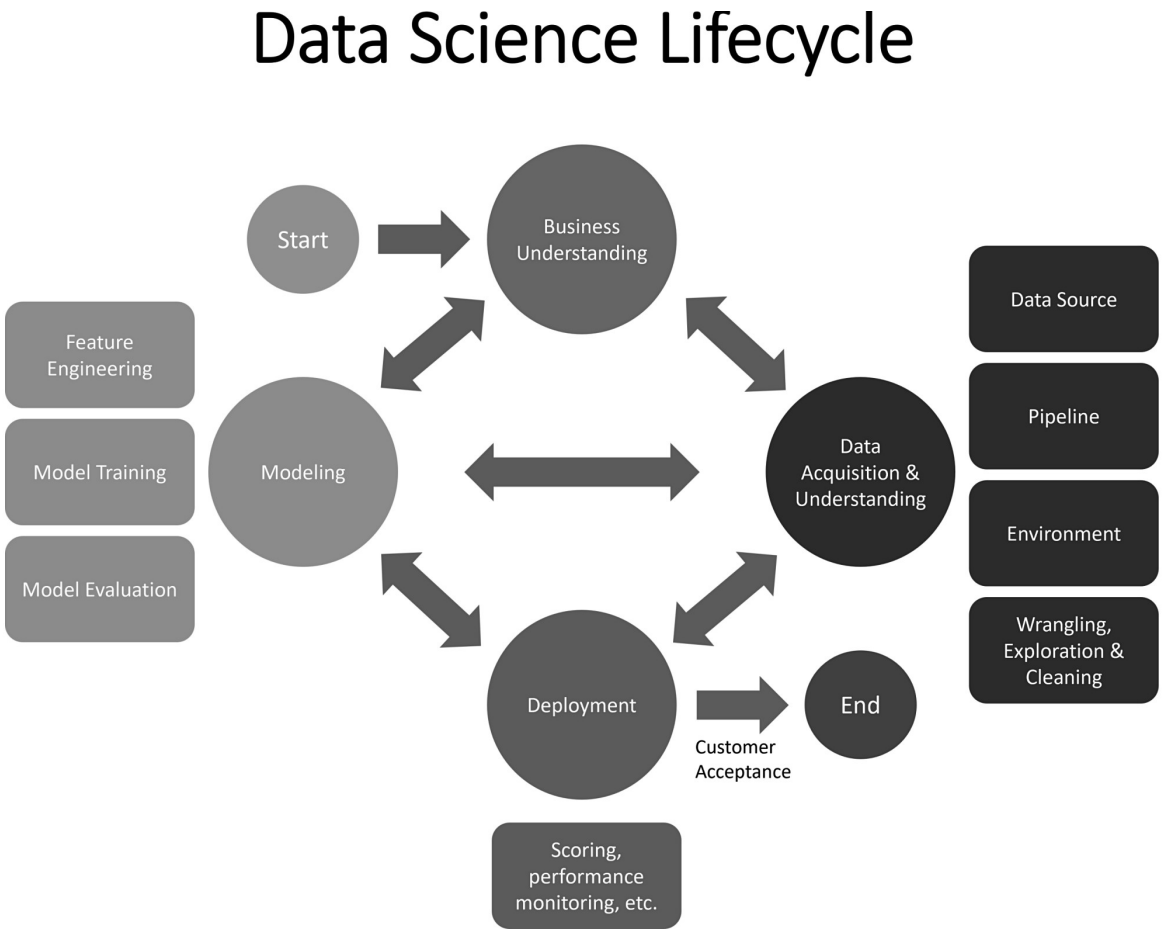

Figure 1.2 – The TDSP lifecycle

At a high level, the TDSP lifecycle outlines the following stages of data science projects:

1. **Business Understanding**: This stage involves working with project stakeholders to assess and identify the business problems that are being addressed by the project, as well as to define the project objectives. It also involves identifying the source data that will be used to answer the business problems that were identified.

2. **Data Acquisition & Understanding**: At this stage, the actual ingestion of data begins, ensuring a clean, high-quality dataset that has a clear relationship with the business problems identified in the **Business Understanding** stage. After having performed initial data ingestion, in this stage, we explore the data to determine whether thedata quality is, in fact, adequate.
3. **Modeling**: After ensuring we have the right data that help address the business problems, we now perform **feature engineering** (**FE**) and model training. By creating the right features from your source data and finding the model that best answers the problem specified in the **Business Understanding** stage, in this stage we determine the model that is best suited for production use.
4. **Deployment**: This is where we operationalize the model that was identified in the *Modeling* stage. We build a data pipeline, deploy the model to production, and prepare the interfaces that allow model consumption from external applications.
5. **Customer Acceptance**: By now, we have a data pipeline in place and a model that helps address the business challenges identified at the beginning of our project. At the **Customer Acceptance** stage, we get agreement from the customer that this project in fact helps address our challenges and identify an entity to whom we hand off the project for ongoing management and operations.

For more details about the TDSP, refer to `https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview`.

Tooling and infrastructure

Big data projects will require specialized tools and infrastructure to process data at scale and with low latency. The TDSP provides recommendations for infrastructure and tooling requirements for data science projects. These recommendations will include the underlying storage systems used to store data, the analytical engines (such as SQL and Apache Spark), cloud services to host ML models, and more.

Azure Synapse offers the infrastructure and development tools needed in big data projects from data ingestion through data storage, with the option of analytical engines for data exploration and to serve data to users at scale, as well as modeling, and data visualization. In the next sections, we will explore the full data lifecycle and how Azure Synapse helps individuals deliver E2E advanced analytics and data science projects.

The need for a fast and highly scalable data exploration service

Data warehouses, and SQL-based databases, have reached a level of maturity where the technologies are stable, widely available from a variety of vendors, and popularly adopted by enterprises. *Structured* databases are efficiently stored, and queries are resolved by using techniques such as indexing and materialized views (among other techniques) to quickly retrieve the data requested by the user.

Unstructured data, however, does not have a pre-defined schema, or structure. Storing unstructured data optimally is challenging, as data pages cannot be calculated in advance the way they are in typical SQL databases. The same challenges apply to the processing and querying of unstructured data.

Application logs and IoT device data are good examples of unstructured data that is produced at low latency. They are text-heavy but without pre-defined text sizes. An application log can not only contain clickstreams, user feedback, and error messages, but also dates and device **identifiers** (**IDs**). IoT device data may include facts such as a count of objects scanned and measures, but also barcode numbers, descriptive text, coordinates, and more.

This is all high-value data that companies now realize can be useful to improve products and respond quickly to market changes and user feedback. Therefore, being able to efficiently store, process, query, and maintain unstructured data is a real requirement for companies of all sizes. But managing big data by itself is not enough—we need the means to efficiently acquire, manage, explore, model, and serve data to end users. In short, we need to realize the full data lifecycle to unlock insights and maximize the value of data. On top of that, we need to make sure that your company's data, being such a valuable asset, is well protected from unauthorized access, and that the analytical environment adheres to mission-critical requirements imposed by enterprises. Let us now look at how Azure Synapse helps address these needs.

What is Azure Synapse?

Azure Synapse is a unified analytics platform that brings together several cloud services to help you manage your data science projects from data ingestion all the way to serving data to end users. *Figure 1.3* illustrates the service architecture for Azure Synapse Analytics:

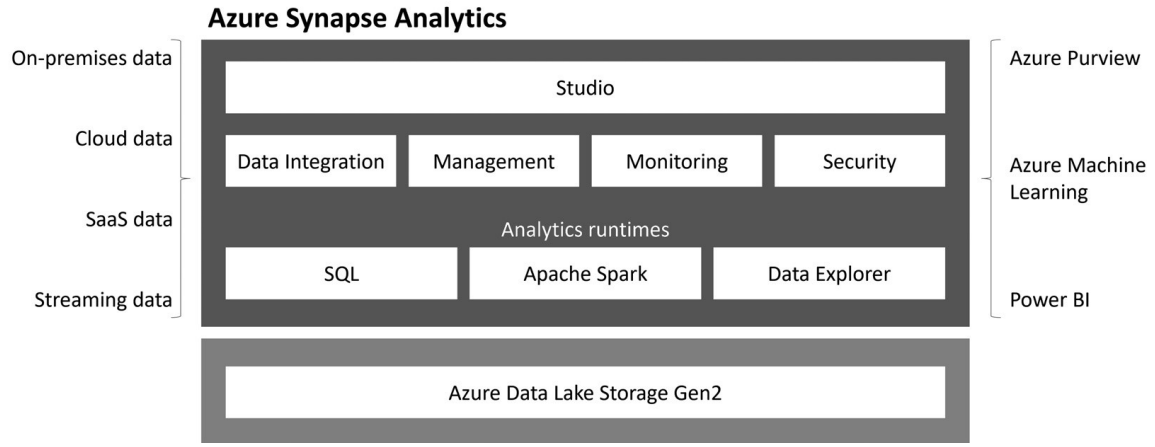

Figure 1.3 – Azure Synapse Analytics service architecture

All these capabilities are managed by the *umbrella* Azure Synapse service in the form of what is called an Azure Synapse *workspace* (the shaded area on top in *Figure 1.3*). When you provision a new Azure Synapse workspace, you are offered a single point of entry and single point of management for all services included in Azure Synapse. You don't need to go to different places to create data ingestion pipelines, explore data using Apache Spark, or author Power BI reports—instead, all the work is done through one development and management environment called **Azure Synapse Studio**, reachable through `https://web.azuresynapse.net`.

Before Azure Synapse, E2E advanced analytics and data science projects were built by putting together several different services that could be hosted on the cloud or on-premises. The promise of Azure Synapse is to offer one platform where all advanced analytics tasks can be performed.

Let's look in detail at the capabilities offered by Azure Synapse.

Data integration

Leveraging the **Azure Data Factory** (**ADF**) code base, Azure Synapse pipelines offer a code-free experience to build data integration jobs that enable data ingestion from data sources in the cloud, on-premises, **Software-as-a-Service** (**SaaS**) sources, data streaming, and more. It includes native connectors to more than 95 data sources.

With Azure Synapse pipelines, data engineers can build E2E workflows for data moving and processing. Azure Synapse supports nested activities, linked services, and execution triggers, and offers common data transformation and data wrangling (transforming data, or mapping data to other columns) activities. In Azure Synapse, you can even add a notebook activity that contains complex logic to process data using an **Azure Synapse notebook**, a code-rich experience, or use flowcharts with a rich user-friendly interface that implements complex pipelines using a code-free experience. This is illustrated in *Figure 1.4*.

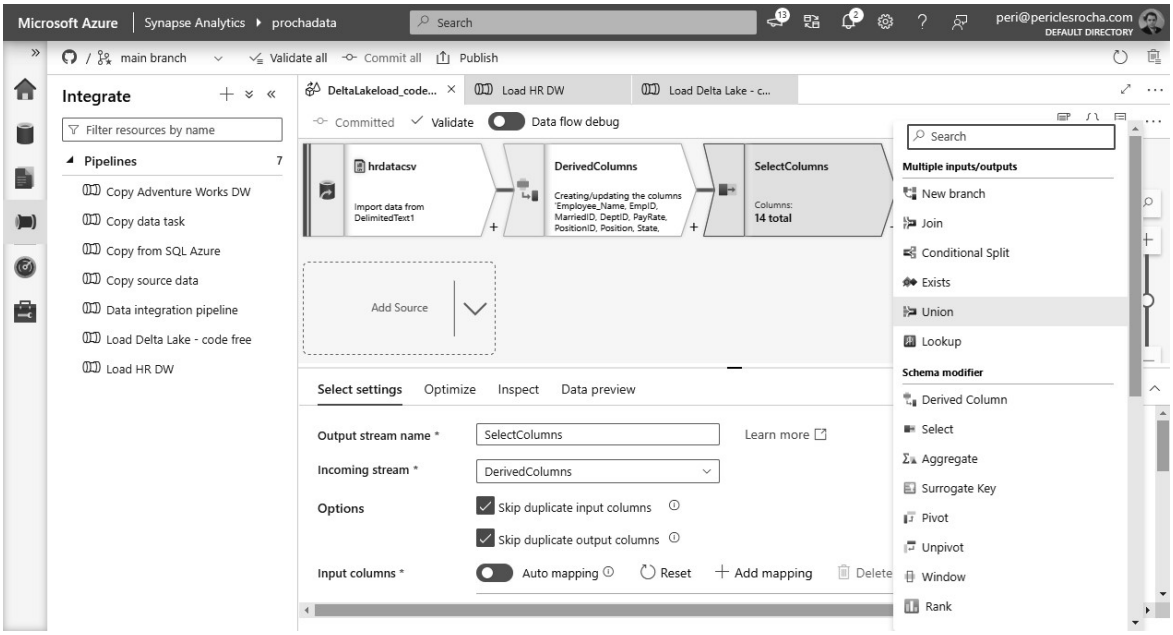

Figure 1.4 – Code-free pipeline and data flow authoring experience

> **Note**
>
> Not all features offered in ADF are available in Azure Synapse Analytics. To explore the differences between ADF and Azure Synapse Analytics, visit `https://docs.microsoft.com/en-us/azure/synapse-analytics/data-integration/concepts-data-factory-differences`.

You should avoid loading data into Azure Synapse SQL using traditional methods that are normal practice in Microsoft SQL Server. For example, issuing batch `INSERT`, `UPDATE`, and `DELETE` statements to load or update data is not an optimal process in Azure Synapse SQL, because the **massively parallel processing** (**MPP**) engine in Azure Synapse SQL was not designed for singleton operations. To help load data efficiently, besides using pipelines, Azure Synapse offers a convenient `COPY` **Transact-SQL** (**T-SQL**) command that helps move data from **Azure Data Lake Storage Gen2** (**ADLS Gen2**) to Synapse SQL tables in an optimal fashion.

Enterprise data warehousing

An enterprise data warehouse—or, simply put, a data warehouse—is a centralized system that integrates data from disparate data sources to enable reporting and analytics in organizations. It stores the data efficiently and is configured to serve data through reporting or user queries, without inflicting overhead on transactional systems.

Azure Synapse offers a highly scalable data warehousing solution through Synapse SQL pools—a distributed query engine to process SQL queries at **petabyte** (**PB**) volume. The SQL analytical engine in Azure Synapse is an evolution of a product previously called Azure SQL Data Warehouse. An Azure Synapse workspace can have several Synapse SQL pools, and the user can run queries using any of the compute pools available. *Figure 1.5* illustrates the ability to pick the desired compute pool for a given query. Pools that have a gray icon without a checkmark are either stopped or not available for use:

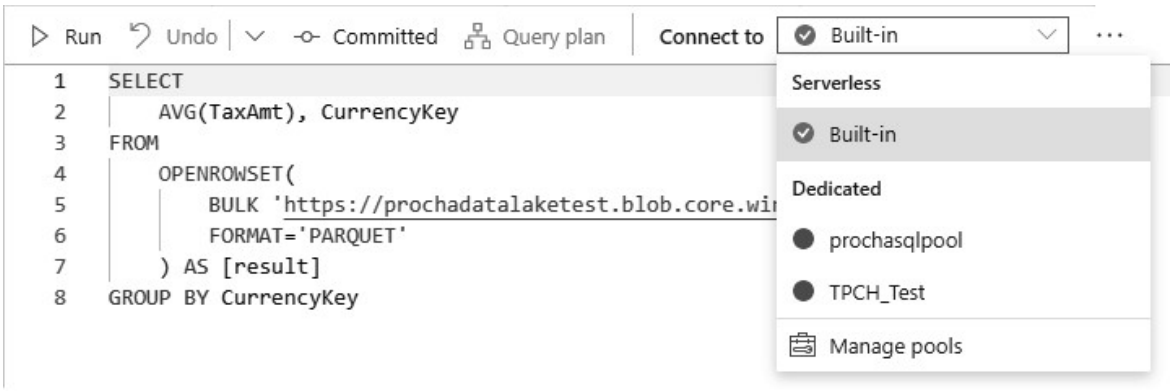

Figure 1.5 – Picking a SQL pool in the query editor

Synapse SQL is offered in two flavors, as detailed here:

- **Dedicated SQL pool**: This is a pre-provisioned compute cluster that offers predictable performance and cost. When a dedicated SQL pool is provisioned, the cluster capacity is reserved and kept online to respond to user queries, unless you choose to pause the SQL pool—a strategy to save money when the cluster is not in use. Dedicated SQL pools run an MPP engine to distribute data across nodes on a cluster and retrieve data. A central **control node** receives user queries and distributes them across the cluster nodes, resolving the user queries in parallel. When you provision a new dedicated SQL pool, you specify its desired cluster size based on your **service-level objective**. Dedicated SQL pool sizes range from one cluster node to up to 60 nodes processing user queries.
- **Serverless SQL pools**: A query engine that is always available to use when needed, mostly applicable for unplanned use, or *bursty* workloads. You do not need to pre-provision serverless SQL pools. You are charged based on the data volume processed in queries. Every Azure Synapse workspace includes a serverless SQL pool. The distributed query processing engine that runs serverless SQL pools is more robust and more complex than the engine that runs dedicated SQL pools and assigns resources to the cluster as needed. You do not control the size of your compute pool or how many resources are allocated to user queries.

Figure 1.6 illustrates the service architecture for dedicated and serverless SQL pools:

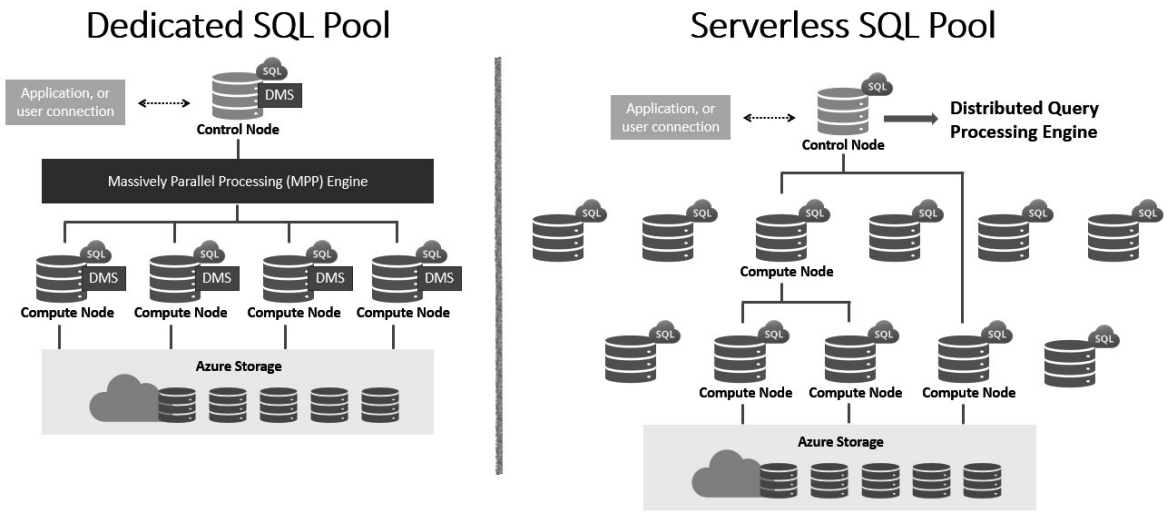

Figure 1.6 – Service architecture for dedicated and serverless SQL pools (adapted from https://docs.microsoft.com/azure/synapse-analytics/sql/overview-architecture)

To learn in depth how serverless pools work in Azure Synapse, I recommend the *POLARIS: The Distributed SQL Engine in Azure Synapse* white paper, which can be found at `https://www.vldb.org/pvldb/vol13/p3204-saborit.pdf`.

Exploration on the data lake

Through native integration with ADLS Gen2, serverless SQL pools allow you to query data directly from files residing on Azure Storage. You can store data in a variety of file formats, such as Parquet or **comma-separated files** (**CSV**), and query it using the familiar T-SQL language.

Exploration on the data lake offers a quick alternative for users who want to explore and experiment with the existing data before it gets loaded into Synapse SQL tables for high-performance querying. In *Figure 1.7*, you can see a T-SQL query that uses the `OPENROWSET` operator to reference data from a Parquet file stored on ADLS:

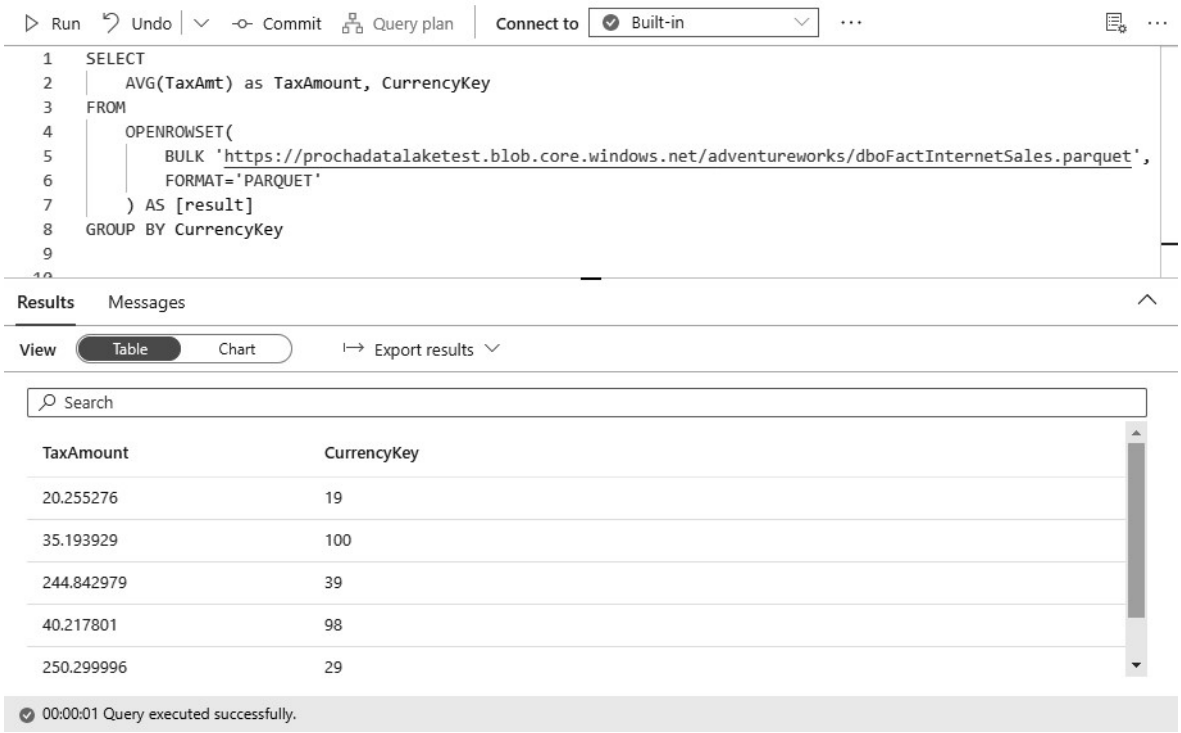

Figure 1.7 – Running T-SQL queries to query data stored on the data lake

This is a powerful capability for users who want to explore data before they decide how to store it to enable the processing of queries at scale. To learn more about exploring data on the data lake using serverless SQL pools, visit `https://learn.microsoft.com/azure/synapse-analytics/get-started-analyze-sql-on-demand`.

Apache Spark

Apache Spark is an open source, highly scalable big data processing engine. It achieves high performance by supporting in-memory data processing and automatically distributing jobs across nodes in a cluster of servers. Apache Spark is widely popular in the data science community not only for its performance benefits (achieved due to its support for in-memory processing and scalability, as described), but also due to the fact that it has built-in support for popular languages such as Python, R, and Scala. Some Apache Spark distributors add additional support for third-party languages as well, such as SQL or C#.

Azure Synapse includes a fully managed Spark service that can be used for data exploration, data engineering, data preparation, and creating ML models and applications. You can choose from a range of programming languages for your data exploration needs, including C#, Scala, R, PySpark, and Spark SQL. The Apache Spark service offered by Azure Synapse is automatically provisioned based on your workload size, so you do not need to worry about managing the actual instances in the cluster.

Apache Spark in Azure Synapse comes with a rich set of libraries, including some of the most used by data engineers and data scientists, such as NumPy, Pandas, Scikit-learn, Matplotlib, and many others. You can also install any packages that are compatible with the Spark distribution used in your Apache Spark pool.

> **Note**
>
> To see a full list of libraries that are pre-installed in Apache Spark pools in Azure Synapse, visit `https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-version-support` and select your desired Spark runtime version.

To explore data using Apache Spark, Azure Synapse offers a notebook experience that allows users to use markdown cells and code cells, as with other popular notebook experiences that are available on the market. This experience is illustrated in *Figure 1.8*.

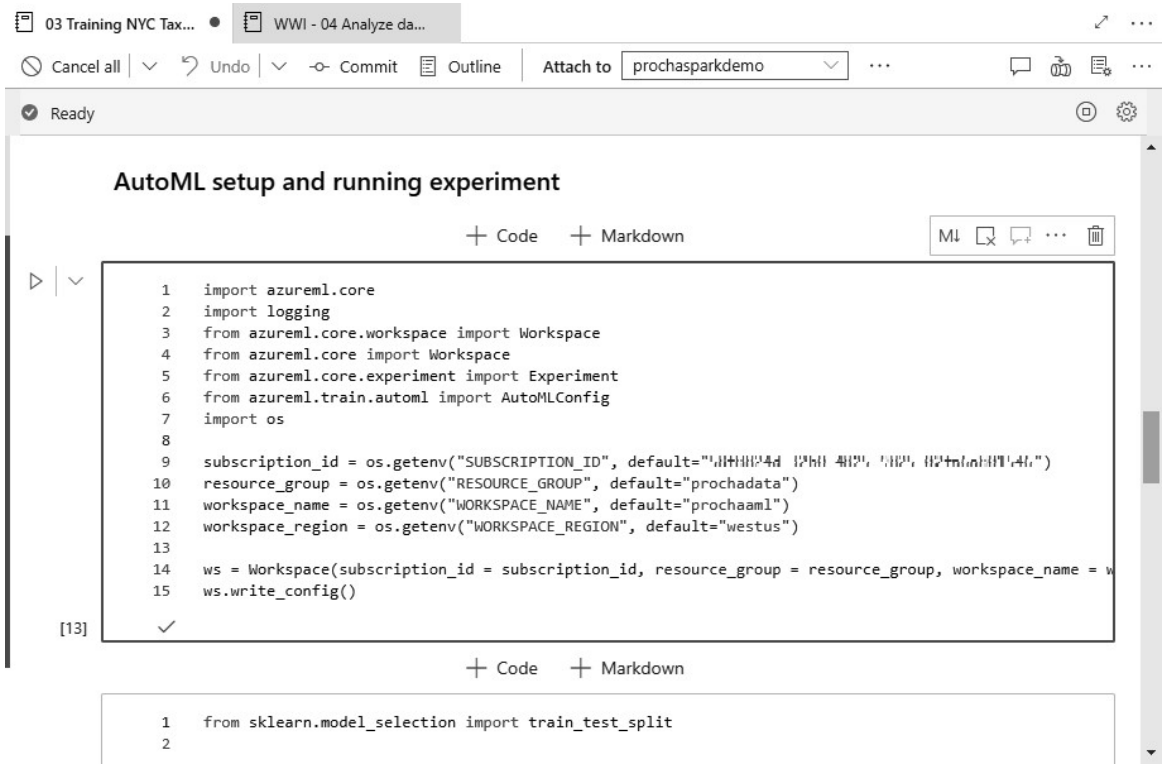

Figure 1.8 – Synapse notebooks authoring experience (subscription ID obfuscated)

Notebooks are saved in your Synapse workspace (or in your source control system if you configured Git integration) just like other workspace artifacts, so anyone connecting to the same workspace will be able to collaborate on your notebooks.

Log and telemetry analytics

Azure Synapse includes native integration with Azure Data Explorer to bring log and telemetry data to E2E advanced analytics and data science projects. You can pre-provision Data Explorer pools in Azure Synapse and have reserved compute capacity for your analytical needs.

Through Azure Synapse Data Explorer (in preview at the time of writing), Data Explorer pools in Azure Synapse enable interesting new scenarios for analysts, data scientists, and data engineers. For example, they offer integration with notebooks in Azure Synapse, allowing you to explore data using your language of choice, in a fully collaborative environment. As you can see in *Figure 1.9*, by right-clicking a table on a Data Explorer pool database and selecting **New notebook**, Azure Synapse Studio can automatically generate a notebook with code to load that table to a Spark data frame:

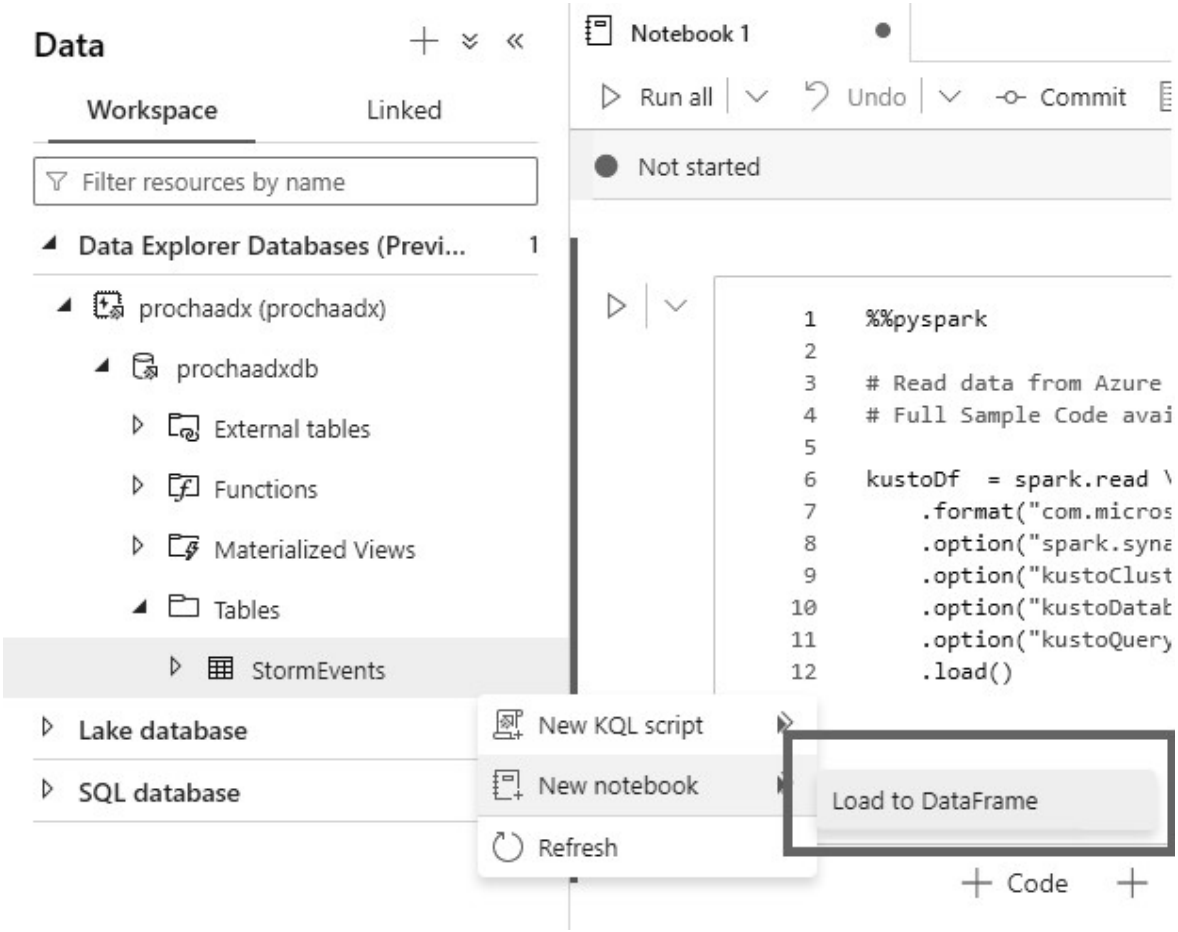

Figure 1.9 – Notebooks reading data from Data Explorer pools

The notebook experience is the same as with Apache Spark reading data from the data lake, except in this case, Azure Synapse generates code for you to load data from Data Explorer into a Spark DataFrame (a table-like data structure that allows you to work with data).

Integrated business intelligence

Having all these data capabilities at your fingertips, it would make sense to be able to richly visualize data. Azure Synapse offers integration with Microsoft Power BI, allowing you to add Power BI datasets, edit reports directly in Azure Synapse Studio, and automatically see those changes reflected in the Power BI workspace that hosts that report.

> **Note**
>
> Azure Synapse does not provision new Power BI workspaces for you. Instead, you add your existing Power BI workspaces to the Azure Synapse workspace using a linked service connection. A separate license to Power BI may be required.

Thanks to the **Azure Active Directory** (**AAD**) integration, connecting to the Power BI service is a simple process. Synapse Studio uses your credentials to look for Power BI workspaces in your AAD tenant and allows you to select the desired one from a combobox, as illustrated in *Figure 1.10*.

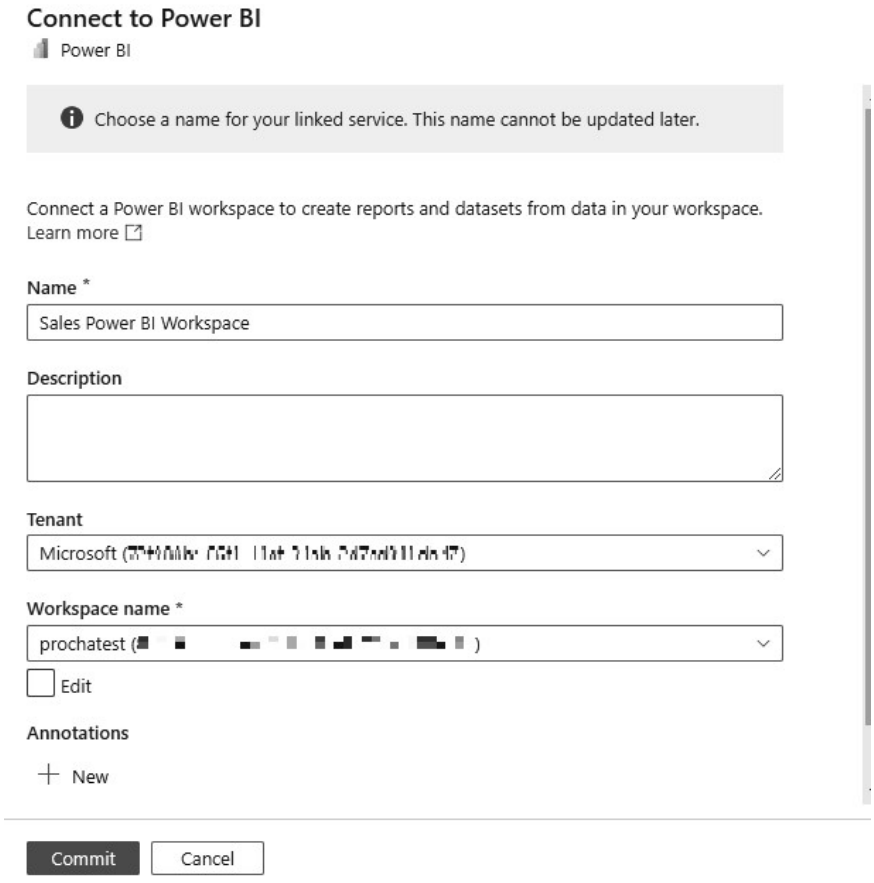

Figure 1.10 – Adding a Power BI workspace as a linked service in Azure Synapse (tenant and workspace globally unique IDs (GUIDs) obfuscated)

This is a powerful and quite useful capability. Not only does it allow you to be more productive and avoid switching between apps to do different work, but it also allows analysts to see the shape and form of their data while they are still exploring it and experimenting with transformations.

Data governance

With the growth of a data culture in corporations, an explosion happened in the number of data marts, data sources, and amount of data that can be used for analytical needs. In Azure Synapse alone, projects normally consume data residing on several data sources. This data can then be copied to a data lake on Azure, transformed, and eventually copied to SQL tables. That is a lot of data and metadata to maintain! How do you get a global view of all the data assets in your organization, and how do you govern this data and classify sensitive data so that it is not misused?

Microsoft Purview is Microsoft's data governance solution for enterprises. It connects to data on-premises, on the cloud, and even to SaaS sources, giving companies a unified view of their data estate. It has advanced data governance features such as data catalogs, data classification, data lineage, data sharing, and more. You can learn more about Microsoft Purview at `https://azure.microsoft.com/en-us/services/purview/`.

> **Note**
>
> Just as with the Power BI integration, Microsoft Purview requires you to have a Purview account, with the appropriate rights, configured separately from your Synapse workspace. To learn how to connect a Synapse workspace to a Purview account, visit `https://docs.microsoft.com/en-us/azure/synapse-analytics/catalog-and-governance/quickstart-connect-azure-purview`.

Configuring your integration with Purview is a simple process: Azure Synapse Studio allows you to pick the Purview account from a list of subscriptions, or to provide the details of your Purview account manually. Once you have configured the integration, you can manage it from Synapse Studio, as illustrated in *Figure 1.11:*

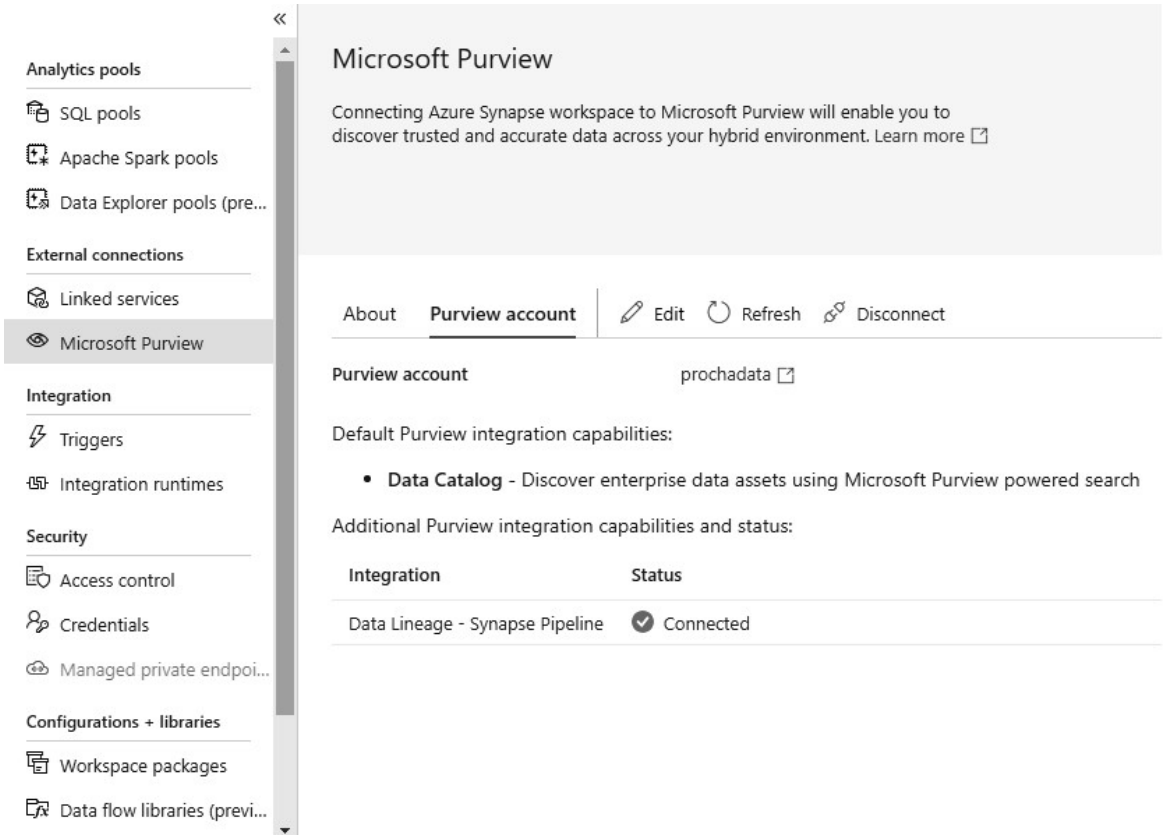

Figure 1.11 – Managing the Microsoft Purview integration

After configuring the Purview integration on a Synapse workspace, you benefit from the following features:

- **Discovery**: Search for any data assets cataloged by Purview by using the global search box.
- **Data lineage**: Understand how the data traveled through the organization and how it was transformed before it landed in the current shape and location. It also allows you to see the raw form of data before it was transformed.
- **Connect to new data**: Having discovered new data assets, instantly connect to them using linked services. From here, you can leverage any service on Azure Synapse to work with the data, from experimentation on Apache Spark to moving data using pipelines.
- **Push lineage back to Microsoft Purview**: After you apply transformations to data and create new datasets on Azure Synapse, you can push metadata that describes your new datasets to Microsoft Purview's central repository for discovery from future users.

While Purview integration is outside of the scope of this book, make sure you understand how to make governance a first-class citizen in your projects—it is a critical aspect of analytical projects and is quickly becoming a hard prerequisite for enterprises.

Broad support for ML

ML is a first-class citizen in Azure Synapse. Models can be trained on Apache Spark, as discussed previously, using a variety of algorithms and libraries, such as Spark MLlib or Scikit-learn. Another option is to connect to the Azure Machine Learning service from within an Azure Synapse notebook and train models using Azure Machine Learning's compute engine. Because Azure Machine Learning offers **automated ML** (**AutoML**), you do not even need to know the best algorithm and features to achieve your objective: AutoML tests a series of parameters and algorithms with the given data and offers you the best algorithm based on a series of results.

Besides model training, Azure Synapse can consume models that were previously trained to run batch scoring with data residing on Azure Synapse. The SQL analytical engine in Azure Synapse can import **Open Neural Network Exchange** (**ONNX**) models (which can be generated from Azure Machine Learning) into its model registry and allow you to use the `PREDICT` function in T-SQL to score columns in real time, as part of regular SQL queries. This is quite powerful!

For example, given a `dbo.Trips` SQL table that contains New York taxi trip data, the query shown in *Figure 1.12* uses the model scored with `Id = 60` to predict taxi fares, given other columns such as passenger count, trip distance, and the date and time of the trip:

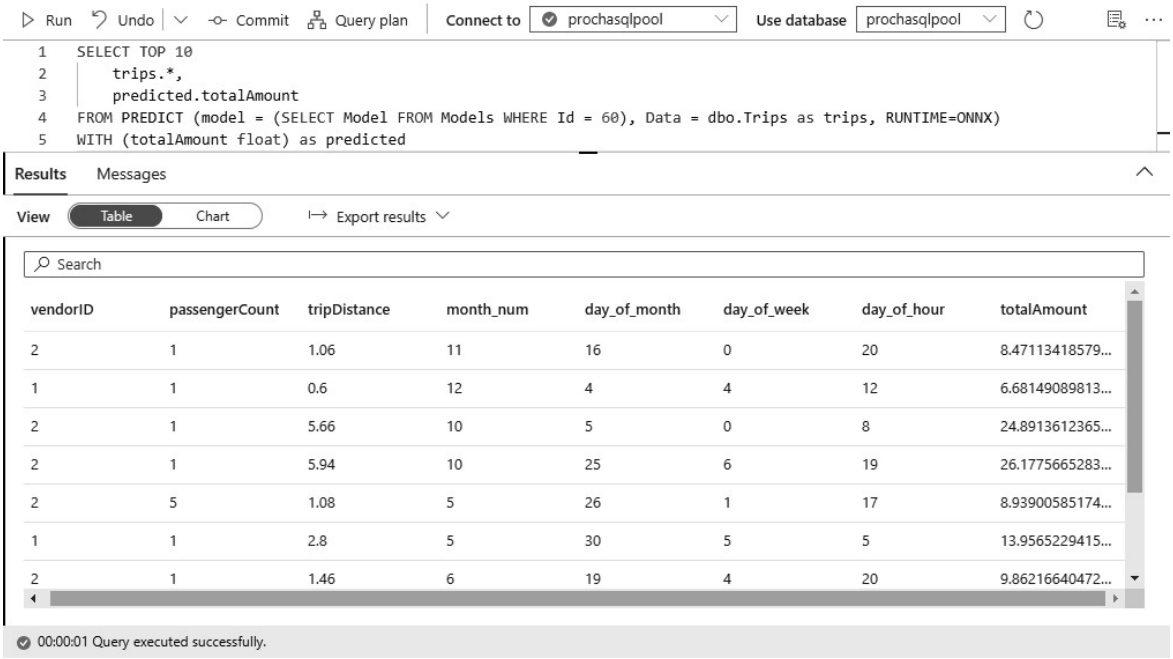

Figure 1.12 – Using PREDICT to score columns in a SQL query

Note that in this example, the model is stored in Azure Synapse SQL's model registry and scoring is done locally, which produces a quick query response time with negligible impact on the query plan. No external services are called.

For consumption of more complex models, or in scenarios where an ML application needs more complex logic that can be better achieved by using a language other than T-SQL, Apache Spark is a perfect alternative. By leveraging Spark pools in Azure Synapse, the regular Notebook experience can also be used for batch scoring.

Security and Managed Virtual Network

The fact that Azure Synapse offers all these cloud services on a single platform may give the impression that it is hard to protect your data. In reality, Azure Synapse workspaces can be created so that they are fully isolated from other workspaces, at the network layer. This is achieved by using **Managed Virtual Network** (or **Managed VNet**) in Azure Synapse.

Managed VNet manages the network isolation, and you do not need to configure **network security group** (**NSG**) rules to allow traffic to and from your virtual network. When associated with a managed private endpoint, workspaces configured on a Managed VNet are fully protected against data exfiltration.

Besides network isolation, Azure Synapse offers an industry-leading set of security features for data protection, **role-based access control** (**RBAC**), different authentication mechanisms, threat protection, and more.

> **Note**
>
> For a detailed view of the security capabilities across all Azure Synapse services, refer to the *Azure Synapse Analytics security* white paper at `https://docs.microsoft.com/en-us/azure/synapse-analytics/guidance/security-white-paper-introduction`.

Management interface

As you can tell by now, Azure Synapse offers several different services that allow you to have a unified platform to build your advanced analytics and data science projects—from data ingestion, all the way to serving data to end users using Power BI. To manage all these services and to build your projects, the primary (and almost only) tool we will use is Azure Synapse Studio.

If you have used ADF before, you will find the **user experience** (**UX**) in Azure Synapse Studio a familiar one. It organizes your work and resources you are using by implementing the concept of hubs (seen on the left-hand side of the **user interface** (**UI**)), as follows:

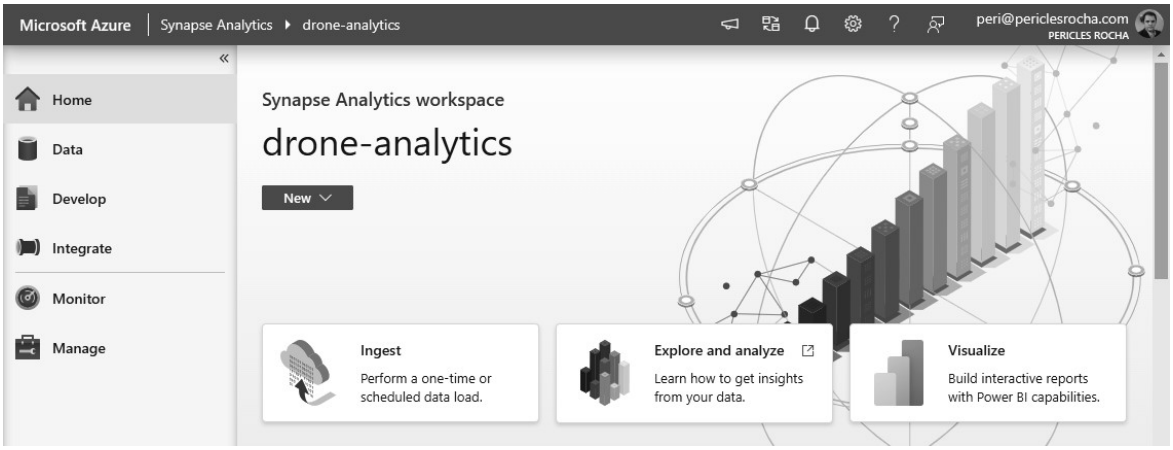

Figure 1.13 – The home page of Azure Synapse Studio

The hubs in Azure Synapse Studio are set out here:

- **Home**: This is the landing page of Azure Synapse Studio. It offers quick links to recently used resources, pinned resources, links to the Knowledge center, and other links to useful content.
- **Data**: The **Data** hub lets you navigate through any SQL, Apache Spark, or Data Explorer pools that you provisioned for your environment, as well as any linked data sources, such as ADLS. It provides a tree view where you can navigate through your compute pools and databases, and even glance at data by right-clicking tables and generating preview scripts.
- **Develop**: Go to the **Develop** hub to find all the SQL/**Kusto Query Language** (**KQL**) scripts you are working on, as well as Synapse notebooks, Power BI reports, and even data flows. All the work you do and *save* (the actual word used in Synapse Studio is *publish*) is stored with the Synapse service, in the context of your workspace. However, you can configure source control and collaboration using Git, and save your work in a Git repository, on Azure DevOps, or on GitHub.
- **Integrate**: This is where you manage all the data integration pipelines created in your Synapse workspace. Azure Synapse Studio provides a code-free experience to build pipelines in a workflow-like environment with a rich UX that mirrors the same experience on ADF.
- **Monitor**: The **Monitor** hub is your **single pane of glass** (**SPOG**) to monitor the status of your compute pools and pipeline runs, Apache Spark jobs, and more. You can also browse through a history of recent activities and verify their execution.
- **Manage**: Finally, the **Manage** hub is where you configure connections to linked services and integration runtimes, scale your SQL, Data Explorer, or Apache Spark pools, and even create pipeline triggers.

Other tools can be used to connect to Azure Synapse services to perform queries and some basic management tasks. Tools such as **SQL Server Management Studio** (**SSMS**) and ADS can be used to run queries on SQL pools. For overall service management and provisioning, while some tasks can be accomplished via T-SQL statements, Azure Synapse can be managed using PowerShell, as well as through the Azure portal.

As you can see, Azure Synapse brings together several different services from data ingestion, through data processing (using your choice of analytical engine) and data visualization to deliver an E2E approach to analytics. It is an industry-defining service offering that brings pieces of the analytical puzzle together for a 360-degree view of your data estate.

This book will focus on Data Explorer in Synapse and how it integrates with these services. So, let's look into it in detail.

What is Azure Synapse Data Explorer?

Before we talk about how Data Explorer is used in Azure Synapse, you may be asking, w*hat is Azure Data Explorer anyways?* Azure Data Explorer is a cloud-based big data platform that enables analytics on large volumes of data, on unstructured, semi-structured, and structured data, with high performance.

Azure Data Explorer comes from a tool built internally at Microsoft for the exploration of telemetry data, which was named *Kusto*. The French explorer Jacques Cousteau inspired the name. The query language it uses is called KQL. Microsoft still extensively uses Azure Data Explorer for telemetry data across its product teams.

At a high level, Azure Data Explorer has the following key features:

- **Data ingestion**: Supports a series of diverse ways to ingest data, from managed pipelines (for example, Event Grid or IoT Hub), connectors and plugins (for example, Kafka Connect or Apache Spark connector), programmatic ingestion through **software development kits** (**SDKs**) or external data loading tools. It supports ingesting up to 200 **MB** of data per second, per cluster node, and load performance responds linearly as you scale the service in and out.
- **Time-series analysis**: Azure Data Explorer is optimized for time-series analysis and processes thousands of time series in a few seconds.
- **Cost-effective queries and storage**: Usage of Azure Data Explorer is charged by compute hours, not by queries, so you can stop your cluster when not in use. It is also charged by storage used. To save on compute hours, Azure Data Explorer supports auto-stop, to automatically stop your cluster after a certain time of inactivity—or you can stop it manually and start again when needed. On storage, Azure Data Explorer offers retention policies, so you can control how long you want to keep your data, also to optimize costs. For long-term storage or cold data, you can always store your data on Azure Storage.
- **Fast read-only query with high concurrency**: Azure Data Explorer is a columnar store and offers fast text indexing. It allows you to retrieve data from a billion records in less than a second.

- **Fully managed and globally available**: You do not need to worry about provisioning hardware, managing operating systems, patching, backup, or even the service infrastructure. Azure Data Explorer is a fully managed **Platform-as-a-Service** (**PaaS**) offering, so you only need to worry about your data. Also, it is globally available, allowing you to provision services closer to where your data is, reducing network latency and respecting data residency.
- **Enables custom solutions**: Azure services such as Azure Monitor, Microsoft Sentinel, and others are built with Azure Data Explorer in their backend. You can leverage the service's REST API and client libraries to build your custom solutions on top of Azure Data Explorer.

> **Note**
> This book explores Azure Synapse Data Explorer, and how it integrates with other Azure Synapse services. To learn more about the standalone service Azure Data Explorer and KQL, a good resource is *Scalable Data Analytics with Azure Data Explorer*, available at `https://www.packtpub.com/product/scalable-data-analytics-with-azure-data-explorer/9781801078542`.

Azure Synapse brings the standalone service Azure Data Explorer to Synapse workspaces, enabling you to complement SQL and Apache Spark pools with an interactive query experience optimized for log and telemetry data. As with dedicated SQL pools, Data Explorer pools are provisioned by you, and compute capacity is reserved while the pool is running. You select your desired cluster size based on your service-level requirements.

As expected, you can use Azure Synapse Studio to run queries on Data Explorer, resume and pause pools, manage the size of your pools by scaling up or down, and view details of your pool such as the instance count, **CPU** utilization, cache utilization, and more.

In Azure Synapse workspaces, when you navigate to the **Develop** hub, you create KQL scripts to explore data on Data Explorer pools. KQL has grown in popularity in recent years due to its adoption by other Azure services, such as Azure Monitor, Microsoft Sentinel, and others.

Integrating Data Explorer pools with other Azure Synapse services

As mentioned previously, before Azure Synapse, data science and advanced analytics projects required engineers to put together several pieces of a puzzle to deliver an E2E solution to users. By bringing Azure Data Explorer natively to Azure Synapse through Data Explorer pools, you no longer need to maintain external connectors and manage services separately. Furthermore, you benefit from the productivity gains of Azure Synapse workspaces, building everything they need on Azure Synapse Studio.

Data Explorer pools on Synapse workspaces offer several benefits, as detailed next.

Query experience integrated into Azure Synapse Studio's query editor

You can query Data Explorer pools using the same tools and the same look and feel you experience with dedicated or serverless SQL pools. Additionally, you can go back and forth between a KQL query on a Data Explorer pool and a T-SQL query on a dedicated SQL pool to get the full context of your data, without having to switch browser tabs or different applications, enabling data correlation across all data sources. Finally, all your KQL scripts can be saved with your SQL scripts and Synapse notebooks into your workspace for future use (or merged into the Git source control mechanism of your choice). In *Figure 1.14*, you can see the **Develop** hub bringing together all your scripts, notebooks, data flows, and Power BI reports:

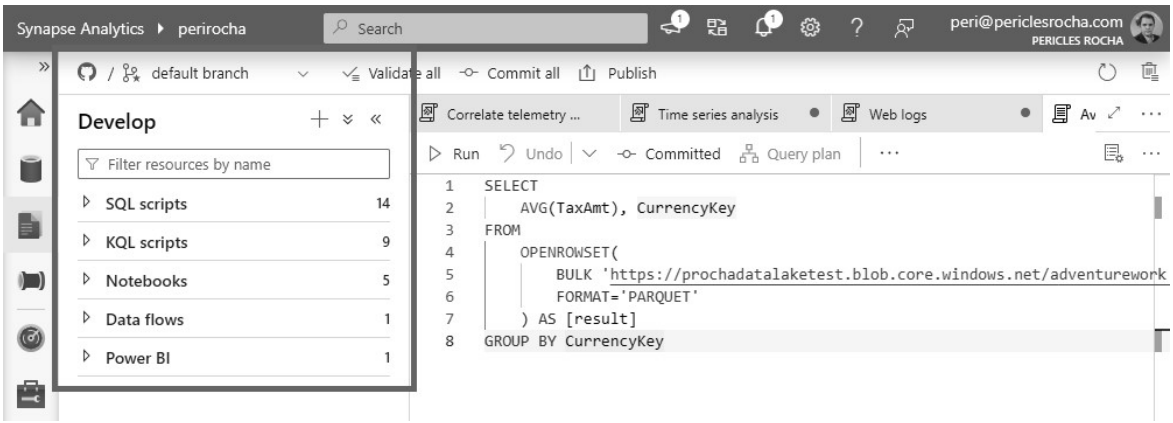

Figure 1.14 – Integrated authoring experience for all your Azure Synapse assets, with source control

> **Note**
>
> Azure Synapse exposes an endpoint for Data Explorer pools the same way as the standalone service Azure Data Explorer. You can still use Azure Data Explorer query tools such as Kusto.Explorer, the Azure Data Explorer web UI, and even the Kusto **command-line interface** (**CLI**) to perform queries if you wish to use them.

Exploring, preparing, and modeling data with Apache Spark

As discussed previously, you can simply right-click a table on a Data Explorer pool and quickly start a new Synapse notebook to use your programming language of choice for data exploration and preparation and to train (and consume!) ML models leveraging Apache Spark. Therefore, you can leverage other benefits of Apache Spark in Synapse, such as Azure Machine Learning integration, and use services such as AutoML.

Data ingestion made easy with pipelines

Among the diverse ways you can load data into Data Explorer pools, as you would expect, Synapse pipelines offer full, native support for the service. If you have existing pipelines and data flows, incorporating Data Explorer pools into your workflows is a simple task.

Unified management experience

Having a SPOG to manage and monitor your services is a huge productivity gain. From Azure Synapse Studio, you can create, delete, pause, resume, and scale Data Explorer pools up or down. You can also monitor the health of pools. Finally, you can control security and access-control rules. When managing settings for your Synapse workspace in the Azure portal, you will also find a central location under **Analytics pools** to create, pause, or delete your Data Explorer pools the same way you do it for SQL and Apache Spark pools. This is illustrated in *Figure 1.15*.

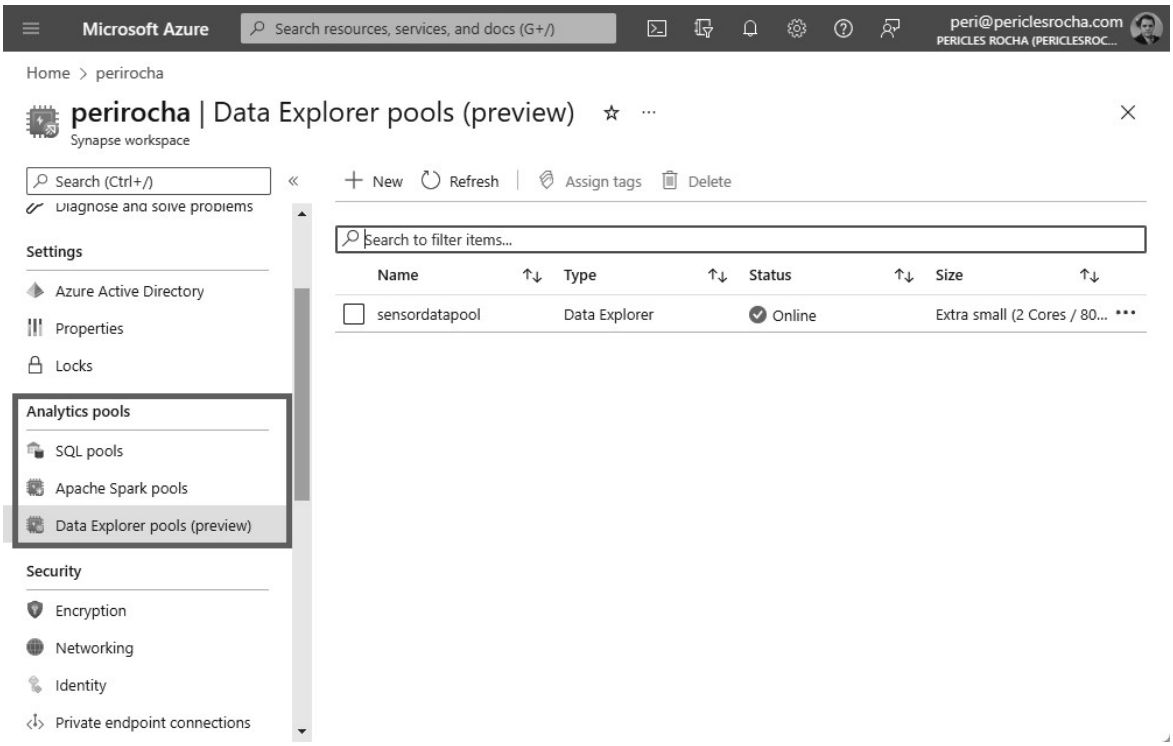

Figure 1.15 – Seamless experience across all analytics pools in the Azure portal

As you can see, Data Explorer is a native service in Azure Synapse and benefits from all the aspects mentioned. It's different from Power BI and Purview in the sense that you don't need to configure it as an external service—instead, Data Explorer pools are like natural cousins of SQL pools and Apache Spark pools, and they share the same experience.

Exploring the Data Explorer pool infrastructure and scalability

Let us look at how Data Explorer pools work behind the curtains.

Any typical deployment of Data Explorer, regardless of being the standalone service or Data Explorer pools in Azure Synapse, will almost always consist of two major services working together, as follows:

- **The Engine service**: Serves user queries, processes data ingestion, and accepts control commands that create or change databases, tables, or other metadata objects (a.k.a. **data definition language** (**DDL**) for seasoned SQL users).
- **The Data Management service**: Connects the Engine service with data pipelines, orchestrates and maintains data ingestion processes, and manages data purging tasks (a.k.a. data grooming) that run on the Engine nodes of the cluster.

These services are deployed through **virtual machines** (**VMs**) in Microsoft Azure, building a cluster of Data Explorer **compute nodes**. These nodes perform different tasks in the architecture of the Data Explorer pool, which we will discuss next.

Data Explorer pool architecture

The Engine service is the most important component in the architecture of Data Explorer pools. There are four types of cluster nodes defined by their respective roles supporting the Engine service, as follows:

- **Admin node**: This node maintains and performs all metadata transactions across the cluster.
- **Query Head node**: When users submit queries, the Query Head node builds a distributed query plan and orchestrates query execution across the Data nodes in the cluster. It holds a read-only copy of the cluster metadata to make decisions for optimal query performance.
- **Data node**: As the *worker bee* in the cluster, it receives part of the distributed query from the Query Head node and executes that portion of the query to retrieve the data that it holds. Data shards are cached in the Data nodes. These nodes also create new data shards when new data is ingested into the database.
- **Gateway node**: Acts as a broker for the Data Explorer REST API. It receives control commands and dispatches them to the Admin node, and sends any user queries it receives to a Query Head node. It is also responsible for authenticating clients that connect to the service via external API calls.

You do not need to worry about how many nodes of which types your cluster contains. The actual implementation of the cluster is transparent to the end user, and you don't have control over the individual nodes.

Scalability of compute resources

Data Explorer was designed to scale vertically and horizontally to achieve companies' requirements, and to accommodate periodical changes in demand. By scaling vertically, you are adding or removing CPU, cache, or **RAM size** for each node in the cluster. By scaling horizontally, you are adding more instances of the specified node size to the cluster. For example, you can configure your Data Explorer pool to start with two instances with eight cores each, and then scale your environment horizontally or vertically as needed.

> **Note**
>
> You cannot control the specific number of CPUs, amount of RAM, or cache size for the VMs used in your clusters. Azure Synapse Data Explorer has a pre-defined set of VM sizes from **Extra Small** (two cores) to **Large** (16 cores) to choose from. These VM sizes have a balanced amount of each compute resource.

Sometimes, it is hard to anticipate how much of a compute resource you will need for a given task throughout the day. Furthermore, if you have high usage of your analytics environment at one point in time during the day but less usage at separate times, you would want to adjust the service to scale automatically as users demand more and fewer resources.

Data Explorer allows you to do just that through **Optimized autoscale**: just set the minimum number of instances you want to have running at any given time of the day, and a maximum number of instances the service can provision in case there's more user demand than the currently allocated resources can support, and Data Explorer pools will scale in and out automatically. So, if your cluster is underutilized, Data Explorer will scale in to lower your cost (while scaling out if the cluster is overutilized). This can be configured in the Azure portal or in Azure Synapse Studio, as seen in *Figure 1.16*.

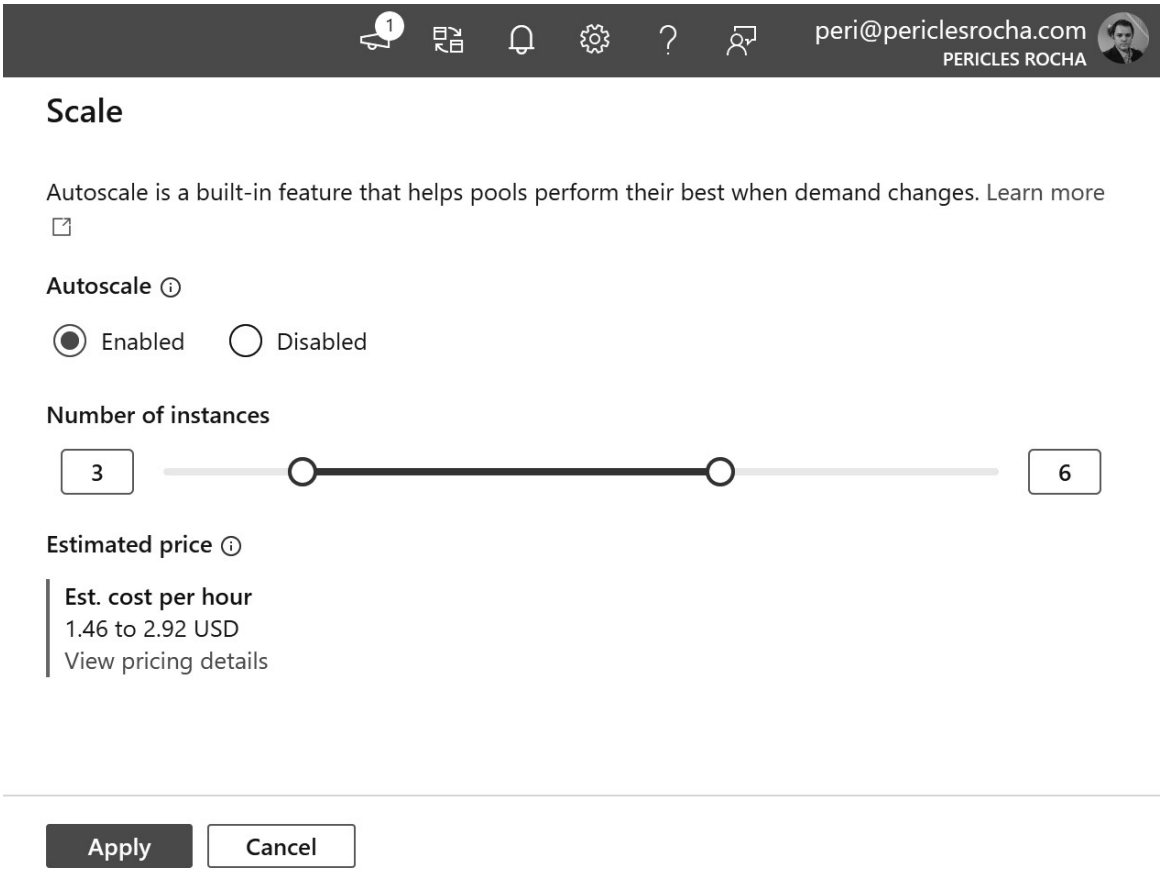

Figure 1.16 – Specifying a minimum and maximum number of instances on Autoscale

The maximum number of instances seen on the slider in *Figure 1.16* scales with the workload size that you selected. With a large compute cluster, you can scale to up to 1,000 instances.

Managing data on distributed clusters

Scaling in and out is great, but you must be thinking: how about my data? The architecture of Data Explorer decouples the storage layer from the compute layer, meaning these layers can scale independently. If more storage is needed, then more resources are allocated to the storage layer. If more compute is needed, your compute VMs will increase in size, or you may have more instances of them.

The Data Explorer service implements database sharding to distribute data across its storage. The Engine service has awareness of each data shard and distributes queries across them. For almost all cases, you don't need to know details about the physical data shards themselves, as Data Explorer exposes data simply through logical tables.

Data is physically persisted in storage, but to deliver a fast query experience, Data Explorer pools cache data in **solid-state drives** (**SSDs**). We will look at how you can define caching policies to balance costs and performance in *Chapter 10*, *System Monitoring and Diagnostics*.

Data shards are distributed across Data nodes using a hash function, which makes the process deterministic—using this hash function, the cluster can determine at any time which Data node is the preferred one for a certain shard. When you scale a Data Explorer pool in or out, the cluster then redistributes the data shards equally across the Data nodes available.

Mission-critical infrastructure

For enterprises, it is not enough to be able to store large amounts of data and retrieve them quickly. Data is a critical asset for companies, and the infrastructure that holds their data needs to be bulletproof to protect it from security and availability challenges and needs to offer developer productivity and sophisticated tooling for monitoring.

Data Explorer pools inherit several of the mission-critical features in Azure Synapse Analytics (and some of these were described in the *What is Azure Synapse?* section of this chapter). Let us look at other features that it offers that are relevant to building mission-critical environments, as follows:

- **AAD integration**: AAD is Microsoft's cloud-based **identity and access management** (**IAM**) service for the enterprise. It helps users sign in to a corporate network and access resources in thousands of SaaS applications such as Microsoft Office 365, Azure services, and third-party applications built with support for AAD.
- **Azure Policy support**: This allows companies to enforce standards and evaluate compliance with services provisioned by users. For Data Explorer, you can use policies such as forcing Data Explorer encryption at rest using a customer-managed key, or force-enabling double encryption, among other policies.
- **Purging of personal data**: Companies have a responsibility to protect customer data, and the ability to delete personal data from the service is a strong asset to help them satisfy the **General Data Protection Regulation's** (**GDPR's**) obligation. Data Explorer supports purging individual records, the purging of an entire table, or the purging of records in materialized views. This operation permanently deletes data and is irreversible.
- **Azure Availability Zones**: Built for **business continuity and disaster recovery** (**BCDR**), Azure Availability Zones replicate your data and services to at least three different data centers in an Azure region. Your data residency is still respected, but in the case of a local failure on a region's data center, your application will fail over to one of the copies in a different data center, but on the same Azure region.
- **Integrated with Azure Monitor**: Collect and analyze telemetry data from your Data Explorer pools to understand cluster metrics and track query, data ingestion, and data export operations performance.

- **Globally available**: At the time of this writing, Azure was available in more than 60 regions worldwide, and the list of regions continues to grow every year. This allows organizations to deploy applications closer to their users to reduce latency and offer more resiliency and recovery options, but also respect data residency rules. For an updated list of Azure regions, visit `https://azure.microsoft.com/explore/global-infrastructure/`.

> **Note**
>
> Not every Azure service is available in every Azure region. For a detailed view of Azure service availability per Azure region, use the *Products available by region* tool at `https://azure.microsoft.com/en-us/global-infrastructure/services/`.

How much scale can Data Explorer handle?

As of July 2022, Microsoft claimed the following usage statistics for Azure Data Explorer globally:

- 115 PB of data ingested daily
- 2.5 billion queries daily
- 8.1 **exabytes** (**EB**) in total data size
- 2.4 million VM cores running at any given time
- More than 350,000 KQL developers

These are important numbers for a managed service. What is even more impressive is that Microsoft claims those numbers are growing close to 100% year over year.

All the details mentioned here about the service architecture and scalability are characteristics of the standalone Azure Data Explorer service too. There are a few special things about Data Explorer in Azure Synapse, so let's explore that next (no pun intended).

What makes Azure Synapse Data Explorer unique?

Even though the underlying service of Data Explorer pools in Azure Synapse is the same as Azure Data Explorer, some capabilities are available exclusively in Azure Synapse. Let us investigate those differences, as follows:

- **Firewall**: Azure Synapse workspaces include a firewall and allow you to configure **IP** firewall rules to grant or deny access to a workspace. This is not available in the standalone service.
- **Availability Zones**: Enabled by default for Azure Synapse workspaces where Availability Zones are available. This can optionally be enabled when using Azure Data Explorer alone.

- **VM sizes for compute**: Azure Data Explorer offers more than 20 different VM configurations to choose from. For Azure Synapse Data Explorer, a simplified subset of the VM configurations is offered, ranging from extra small (two cores) to large (16 cores).
- **Code control**: As previously mentioned, in Azure Synapse you can connect your workspace with a Git repository, Azure DevOps, or GitHub. This option is not available when using the standalone service.
- **Pricing**: For a Azure Synapse workspace, Data Explorer pools pricing is simplified to two meters: **VCore** and **Storage**. When using Azure Data Explorer as a standalone service, you may be charged by using multiple meters such as **Compute**, **Storage**, **Networking**, and the Azure Data Explorer IP markup, which is applied when you make use of fast data ingestion, caching, querying, and management capabilities. Additionally, **Reserved Instances**, which offer discounted prices when you make a commitment to use an Azure service for a certain period (typically 1 or 3 years), are only available for the standalone service Azure Data Explorer, and not for Azure Synapse Data Explorer.

As seen from the preceding points, there is no significant loss of functionality by using Azure Synapse Data Explorer when compared to the standalone service Azure Data Explorer. Azure Synapse Data Explorer includes the benefits seen on the standalone service, and it also incorporates the enterprise features offered with Synapse workspaces. However, is Azure Synapse Data Explorer the solution to every analytical problem? In the next section, you will find out how to decide whether you need Data Explorer pools or not.

When to use Azure Synapse Data Explorer

By now, you should already understand that Azure Synapse Data Explorer is an analytical engine to process queries on unstructured, semi-structured, and structured data, with exceptionally large data volumes, low-latency ingestion, and blazing-fast queries. Data Explorer is not, however, the solution to every data problem. In some cases, you will be better off with a different solution. Let us look at some of the most common analytics scenarios and the most appropriate analytical store in each case, as follows:

- **Scenario**: *I need a classic data warehouse.*

  Recommendation: Do not use Azure Synapse Data Explorer. Use dedicated SQL pools in Azure Synapse, which are optimized for user queries in a typical star schema, even at large data volumes.

- **Scenario**: *My solution requires frequent updates on individual records, and singleton INSERT, UPDATE, and DELETE operations.*

  Recommendation: Do not use Azure Synapse Data Explorer. In such cases, a transactional, operational database will be a better solution. Consider options such as Azure SQL, SQL Server (on-premises, or in an Azure VM), MySQL, or even Cosmos DB for NoSQL scenarios.

- **Scenario**: *My solution needs to run on a cloud other than Microsoft Azure, or on-premises.*

  Recommendation: Do not use Azure Synapse Data Explorer, as it runs exclusively on Azure.

- **Scenario**: *My data demands constant transformation and long-running extract, transform, load (ETL)/extract, load, transform (ELT) processes.*

  Recommendation: Do not use Azure Synapse Data Explorer. Even though you have Synapse pipelines in your Synapse workspace, and you can constantly ingest data into Data Explorer pools, the core scenario for Data Explorer is to offer interactive analytics on big data. You are better off running your ETL/ELT pipelines on Azure Synapse pipelines, ADF, Apache Spark, or even Azure Batch.

- **Scenario**: *I need to train large ML models several times throughout the day.*

  Recommendation: This may be a good scenario for Azure Synapse Data Explorer. In this case, you can prepare data or train models on Apache Spark for Azure Synapse, but note that you will miss out on the real-time characteristic of data analysis that Data Explorer offers. Ideally, you want to use Data Explorer with data streaming from devices and applications in real time, but this still can be a valuable scenario for Azure Synapse Data Explorer. This may be less valuable when using the standalone service Azure Data Explorer, as it will not benefit from the native, in-product integration with Apache Spark (even though a connector for Spark is available for Azure Data Explorer uses).

- **Scenario**: *I have a very small amount of data to analyze.*

  Recommendation: It depends. If your analysis requires a full-text search or **JSON** documents, you may benefit from the indexing capabilities of Azure Synapse Data Explorer. It can also be a suitable alternative if you need to correlate this data with other data stored on Synapse SQL or in the data lake. If you are on a low budget and don't need the added benefit of Azure Synapse, you may be better served with SQL Server, Azure Cognitive Search, or even Cosmos DB.

- **Scenario**: *I need to perform time-series analysis on metric data from sensors, social media, websites, financial transactions, or other fast streaming data.*

  Recommendation: You should use Azure Synapse Data Explorer. Data Explorer pools are optimized for application log and IoT device data and can ingest data at high volumes offering insights in near real time.

- **Scenario**: *I have data in a diverse schema, and with high volumes of data in near real time.*

  Recommendation: You should use Azure Synapse Data Explorer. Data Explorer pools are optimized for unstructured, semi-structured, and structured data and allow you to run interactive analytics on data of any shape.

- **Scenario**: I need to correlate application logs or telemetry data from IoT devices with data sitting in a data warehouse and the data lake.

  Recommendation: You should use Azure Synapse Data Explorer. By leveraging the SQL analytical pools in Azure Synapse (dedicated and serverless), you can use one tool to query all your data, regardless of the analytical store that holds it.

The rule of thumb is to think about Data Explorer pools when you are managing telemetry or log analytics data at scale. You should use it with Azure Synapse when you need to combine your analysis with data from other sources or use the added benefits of Azure Synapse in your project.

Summary

Azure Synapse Data Explorer brings all the innovation that was built into the standalone service Azure Data Explorer into Azure Synapse. By using Data Explorer pools in Azure Synapse, you can correlate data from several different sources, residing on different analytical engines, to get a 360-degree view of all your data and unlock insights.

In this chapter, you learned about the lifecycle of data, the TDSP, and how Data Explorer fits into the analytics landscape. We explored the key components of Azure Synapse and Azure Synapse Data Explorer, and how Data Explorer pools benefit from integration with other Azure Synapse services such as Apache Spark.

Next, you learned about the infrastructure of Data Explorer pools and how they deliver massive scalability. We looked at the service architecture and how Data Explorer pools manage data in a distributed cluster. We also explored the mission-critical features of Azure Synapse Data Explorer that give the trust enterprises need to adopt the solution.

Finally, we discussed what makes Azure Synapse Data Explorer unique when compared to the standalone service Azure Data Explorer, and how to determine whether Azure Synapse Data Explorer is the right solution for you.

In the next chapter, we will explore how to create an Azure Synapse workspace and a Data Explorer pool.