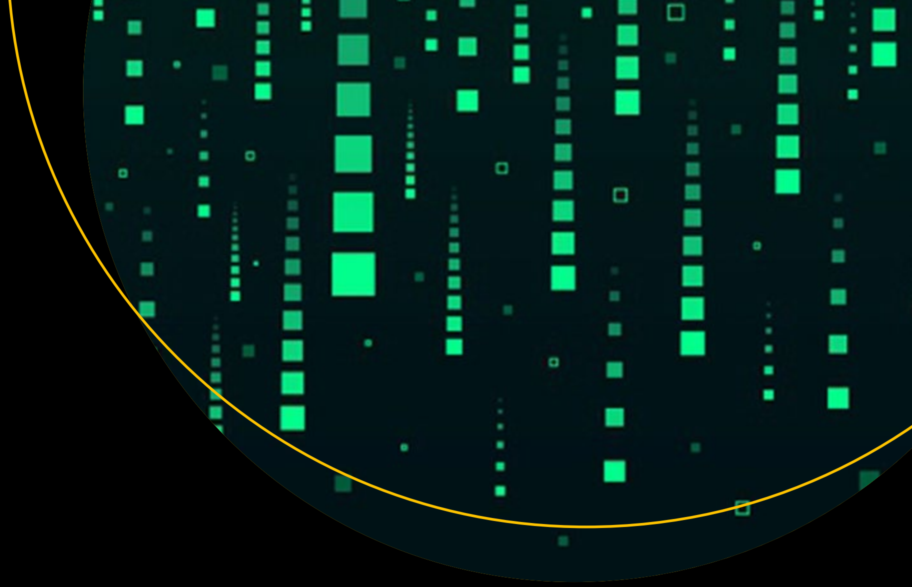


The Modern Data Warehouse in Azure

Building with Speed and Agility on Microsoft's Cloud Platform

Matt How

Apress®

The Modern Data Warehouse in Azure

Building with Speed and Agility on Microsoft's Cloud Platform

Matt How

Apress®

The Modern Data Warehouse in Azure: Building with Speed and Agility on Microsoft's Cloud Platform

Matt How
Alton, UK

ISBN-13 (pbk): 978-1-4842-5822-4 ISBN-13 (electronic): 978-1-4842-5823-1
https://doi.org/10.1007/978-1-4842-5823-1

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

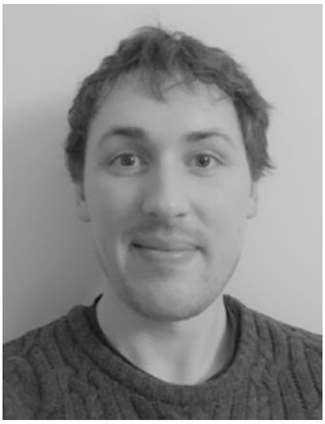
Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484258224. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

To my wife Amy and our children,
for the continual love and support.

Table of Contents

About the Author



Matt How is a professional consultant and international conference speaker who is passionate about data, analytics, and automation. Having spoken at several large conferences across the world, he is committed to sharing knowledge and insight to the wider community. Specializing in the design and delivery of modern data warehouse solutions using the Microsoft Azure Platform, Matt focuses on simplicity and resilience above all when designing cloud solutions. With a growing focus on data science, Matt is now researching techniques to integrate artificial intelligence capabilities into the modern data warehouse at scale.

About the Technical Reviewer

Carsten Thomsen is a back-end developer primarily but working with smaller front-end bits as well. He has authored and reviewed a number of books and created numerous Microsoft learning courses, all to do with software development. He works as a freelancer/contractor in various countries in Europe, using Azure, Visual Studio, Azure DevOps, and GitHub as some of the tools. Being an exceptional troubleshooter, asking the right questions, including the less logical ones, in a most logical to least logical fashion, he also enjoys working with architecture, research, analysis, development, testing, and bug fixing. He is a very good communicator with great mentoring and team-lead skills, and great skills researching and presenting new material.

Acknowledgments

Writing a book was much harder than I ever imagined and so I must start by thanking my awesome wife, Amy, for her everlasting encouragement and support. She always kept the big dreams alive while ensuring my aspirations were founded in reality. I also want to thank my children for being the most welcome distraction to advanced modern analytics I could ever have dreamed of.

Thank you to my parents, friends, brothers, and other family members for their continued interest and encouragement. I sincerely hope they all enjoy receiving the same Christmas gift this year.

I want to acknowledge and thank all my colleagues at Adatis, many of whom have been an excellent sounding board for many of the concepts and ideas included in this book. A special thanks to the directors for their support and guidance throughout the process; they have always been exceptionally accommodating of both personal and professional achievements.

Prior to this project, I had never considered authoring a book and so I am sincerely grateful to Jonathan at Apress for reaching out to me and sparking the initial conversation. His continued guidance and patience have been a true blessing. In addition, I want to thank Jill for keeping everything on track and Laura for her sage advice throughout the editing process.

Finally, I want to thank Carsten for an excellent eye for detail and for providing an abundance of helpful comments and tips as part of his edit. I am very glad to have someone of his experience play a part on the production of this book.

Introduction

An enterprise data warehouse (EDW) is a common, business-critical system that benefits from highly mature concepts and design best practices. In the market today, there is a wealth of books on the topic, some of which examine the differences between the two fundamental ideologies behind the warehouse design, those of Ralph Kimball and his drive for denormalized star schemas and Bill Inmon with his preference for a normalized corporate data warehouse. Others may focus on specific patterns or techniques to solve more tricky modeling problems. However, few focus on the platform that is being used for the data warehouse. Taking nothing away from these books, the concepts they discuss are still relevant today; however, very few books speak specifically about a cloud-based implementation of a data warehouse and how the tooling is different, how the patterns change, and how a developer needs to adapt to the new environment.

Gone are the days when a data warehouse project was a slow-moving, inflexible venture that was difficult to maintain and impossible to extend. We now have an impressive set of tools that allow us to surface analytical insight at massive scale and at incredible speed, without the overhead of maintaining a gigantic server. Not only is a cloud platform perfectly tailored for data processing, but the processes to feed that platform can be completely automated and integrated to just about any source system, making maintenance and development simple and enjoyable. Further to all this, we can now fully explore the different ingestion architectures that comprise streaming, event-based, and batch loading, allowing developers to break free of the “Nightly ETL Window” constraint and fully discover how they can populate the warehouse at the rate of the incoming data.

But is there a reason why an entire book needs to be dedicated to data warehousing in the cloud? Doesn’t the cloud provide the same technology as on-premises just without the server management? The short answer is no. As you go through this book, the hope is that you will discover the nature by which the cloud completely changes the way a data warehouse is built and why it is important to consider making this move. The core concepts of on-premises data warehousing still very much apply, but the way in which they are implemented has drastically changed. The cloud has revolutionized the way developers can reason about a problem and even eliminated some compromises that had to be made in the years gone by. This is not without cost however; there are new problems to understand and tackle and part of the aim of this book is to talk these issues through and make clear the patterns that solve those issues.

In this book, you will not find much discussion of Online Transaction Processing (OLTP) type systems nor of the wider capabilities of the Microsoft Azure data platform. This book will not discuss why you should implement either Kimball or Inmon or explain how to create a flashy executive level dashboard. Instead this book is a discussion about the key technologies in the Microsoft Azure data platform that lend themselves to data warehousing and how they connect together. I will explain how to choose a SQL engine that is tailored for your analytical requirements, how to create data movement processes that scale, and how to extend your warehouse to become intelligent and modern.

If you are already building SQL data warehouses, you may wonder if you need a book such as this. You know SQL. You know ETL. What can this book tell you that you do not already know? Well, SQL server is changing. And given that Microsoft is a cloud-first company, the newest features and biggest developments are shipped to the Azure versions of SQL months if not years before they hit the box product. Not only this, there are features arriving in the Azure data platform that will NEVER be available in the box product. Things like Accelerated Database Recovery (ADR) simply cannot be implemented on-premises, and if your organization cares about their recovery time objective (RTO) and recovery point objective (RPO), then this is a feature you need to understand. Ultimately there are an increasingly small number of reasons why a company would choose to avoid cloud software and this book hopes to dispel the last of those.

I sincerely hope that this book eradicates any anxiety about making a move to the cloud, and if your organization has embraced the cloud already, then I aim to provide further insight into how the technologies work at a low level and advise on the patterns and architectures that should be utilized to get the most out of them.

Who This Book Is For?

If you are already building on-premises Microsoft SQL Server data warehouses using common tools such as SSIS, then this book will explain how to move that knowledge into the cloud, giving, where possible, comparisons about the way a thing was done in that world and how it should be done in the cloud. If you are already utilizing some of the

Azure data platform, then this book will hopefully provide a better understanding of how each service operates and why it works the way it does. If you are already successfully running and developing data warehouses with Azure Synapse Analytics (formerly Azure SQL Date Warehouse) or Azure SQL Database and Azure Data Factory, then I hope this book will help to solidify your knowledge and perhaps provide some fresh ideas or patterns that you could use in future development.

If you hope to understand the entire Azure data platform, then this book will not be broad enough to answer all your questions. For example, we will not go deeply into Cosmos DB or any of the third-party database offerings in Azure. Additionally, we will not cover off core data modeling concepts other than where this is critical to the implementation of an Azure Synapse Analytics instance. Despite this, a good working knowledge of the other data stores and technologies available in Azure will open up many new avenues for you to explore that can allow for exciting and highly valuable extensions to a traditional data warehouse.

Assumptions About You

The people that will get the most out this book will be already experienced with data warehousing core concepts and the terminology that goes along with it. A good understanding of the common challenges and why they need to be overcome is also a good base to start from. I have made the assumption that you and your company are already fairly comfortable that a cloud-based architecture will suit your business requirements, taking into account security, cost, admin, and so on. As this book is not a full examination of a cloud data platform, often a warehouse sits among many other databases, it has to be assumed that you and your company have the ability to connect to the cloud and create the necessary resources for testing and proof of concept work where needed.

With this in mind, I am aware that readers may arrive at this book from a spectrum of job roles. Some may come from an analysis background looking to develop the back-end of their reports so that they are more scalable, whereas some may be more comfortable with the data engineering concepts and therefore be looking to replicate existing solutions but without the overhead and hassle of server management. Either way this book will certainly help in making clear the concepts that need to be understood in order to create a functioning data warehouse in Azure.

The Scope of This Book

In any IT project, scope is key. You need to know what you are getting, so let me make this abundantly clear what this book is and is not.

This book is

- A guide to cloud data architecture for data warehousing scenarios, implemented using Azure SQL technologies, Azure data lake technologies, and Azure integration technologies
- A guide to ingesting data with Azure Data Factory and developing metadata-driven pipelines
- An introduction to ingestion patterns that can be automated, be driven by metadata, utilize streaming, and make use of data lakes
- A point of reference for good practice around logging, auditing, and resilience regarding the aforementioned technologies
- A guide to developing and using project accelerators to improve the pace of development and ensure consistency across teams

This book is not

- A detailed description of how to conduct automated deployments to an Azure platform.
- A guide to data modeling best practice. There will be some mention of data modeling as this is key to the structure of Azure Synapse Analytics, but this will not be a book on Kimball vs. Inmon modeling.
- A manual for data preparation and cleansing. I will explain where these elements would slot into the process but not give an abundance of material on how to clean and prepare your data.

Throughout this book, there are step-by-step guides to assist you getting to a basic level of usage with a service; however, the book as a whole is not a step-by-step guide to creating a functional modern SQL data warehouse on the Azure platform.

Organization of the Book

This book is laid out so that the most important topics are covered upfront and that the key elements of a cloud data warehouse are well understood before continuing into how the development process can be accelerated and some other more advanced topics. However, at the very start, there are some handy sections that cover initial guidance for using Microsoft Azure such as subscription organization, security, development tools, and a glossary of common terms. For all of the walk-throughs in this book, you will need access to an Azure subscription where you have a relatively high level of permission for things like setting up service principals.

The bulk of the book begins from Chapter 2, "The SQL Engine," and focuses on the choices to be made when designing your modern data warehouse and how that process can be accelerated and improved. The following is a brief summary of the content of each chapter to allow you to skip to the most important discussions if needed:

- **Chapter 2: The SQL Engine.** The goal of this chapter is to make clear the distinction between Azure SQL Database and Azure Synapse Analytics and when one option should be chosen over another. The conclusion of this chapter talks about your type of data and what SQL engine would be best suited.

- **Chapter 3: The Integration Engine.** This chapter introduces Azure Data Factory and explains the key building blocks that make it a first-class cloud integration tool and really the only option for data movement within the Azure platform. Additionally, an example of how to copy data from source to sink is included.

- **Chapter 4: The Ingestion Architecture.** As this will be a modern data warehouse that can cope with a much more varied workload, we can now consider different types of data processing. You will discover how you can capitalize on event-based processing and streaming and the additional complexities these options introduce, as well as the more traditional batch-based loading technique.

- **Chapter 5: The Role of the Data Lake.** A revolution in cloud data storage has been the advent of the data lake. While the data lake is a broad topic, this chapter will relate specifically to its purpose in the data warehousing architecture. Effectively, the data lake is a single access point for an entire organization's varied datasets, be it media, tabular data, backups, and others. This makes it an ideal staging location for the data warehouse and when properly implemented can vastly improve the efficiency of the data warehouse.
- **Chapter 6: The Role of the Data Contract.** A large amount of data warehouse processing can be automated and defined in metadata. Things like file schemas, transformation rules, and processing steps can all be stored as metadata in a database of some kind. Throughout this chapter, you will gain an understanding of how metadata can be used to solve several common problems and how to store, fetch, and implement it.
- **Chapter 7: Logging, Auditing, and Resilience.** A crucial piece of a production warehouse is the monitoring and auditing of the ingestion process and being able to catch and resolve instances of bad or mis-shaped data. The concepts outlined here will likely not be new if you are an experienced data warehouse developer, but the specific implementation covered will tie in closely with the metadata mentioned previously in Chapter 6, "The Role of the Data Contract."
- **Chapter 8: Using Scripting and Automation**. With any Azure resource, scripting and automation can be a great asset to assist with deployment and management. This chapter will expand on some common scripts I often find useful and explain their usage.
- **Chapter 9: Beyond the Modern Data Warehouse**. This chapter will talk about how the modern data warehouse can be extended to support analytical tools and even application data. We will look at integrations with Power BI, Cosmos DB, and Analysis Services, explaining the security and reliability concepts at play and describe best practice and patterns for implementation.

CHAPTER 1

The Rise of the Modern Data Warehouse

A data warehouse is a common and well-understood technology asset that underpins many decision support systems. Whether the warehouse was initially designed to act as a hub for data integration or a base for analytical consistency, many organizations make use of the concepts and technologies that underpin data warehousing.

At one point, the concept of a data warehouse was revolutionary and the two key philosophies on data warehousing, those of Ralph Kimball and Bill Inmon, were new and exciting. However, many decades have passed since this point, and while the philosophies have cross-pollinated, the core design and purpose has stayed very much the same, so much so that many data warehouse developers can move seamlessly from company to company because the data warehouse is such a prevalent design. The only thing that changes is the subject matter. This is very unlike more transactional databases that may be designed very differently to support the specific needs of an application.

As the cloud revolution began, more and more services began to find homes in the cloud and the data warehouse is no exception. A cloud-based environment eliminates many common issues with data warehousing and also offers many new opportunities. First of which is the serverless nature of cloud-based databases. By not having to manage the server environment, patching, the operating system (OS) or upgrades, and others, the development team can really focus just on the data processing that needs to be undertaken. In addition, the architecture itself can be scaled so that businesses pay for what they actually use and not for a service that offers growth room for the next five years. Instead, the size of the system can be tailed and charged at per hour increments so that aggressive cost optimizations can be achieved.

In times gone by, the on-premises architecture of data warehouses meant that there were hard limits on the amount of data that could be stored and the frequency at which that data could be ingested. Further, the tools used to populate an on-premises data

M. How, *The Modern Data Warehouse in Azure*, https://doi.org/10.1007/978-1-4842-5823-1_1

warehouse had limited ability to deal with complex data types or streaming datasets, concepts that are now prevalent in the application landscape that feed data warehouses. Businesses now require these sources to be included in their reports, and so the data warehouse must modernize in order to keep up. At present, Azure provides many tools and services to help overcome these problems, many of which can be integrated directly into what would now be known as a modern data warehouse.

In addition to modernizing the database, the tools that operate, automate, and populate the data warehouse also need to keep up in order for the solution to feel cohesive. This is why Azure offers excellent integration and automation services that can be used in conjunction with the SQL database technologies. These tools mean that more can be achieved with less code and confusion, by creating standard patterns that can be applied generically to a variety of data processing problems. Common menial tasks such as database backups can be completely automated, making the issue of disaster recovery much less of a worry. With the latest features of Azure SQL Database, artificial intelligence is used to recommend and apply tuning alterations and index adjustments to ensure database performance is at its absolute best. This works alongside advanced threat detection which ensures databases hosted in Azure are safer than ever.

Finally, businesses are increasingly interested in big data and data science, concepts that both require processing huge amounts of data at scale and maintaining a good degree of performance. For this reason, data lakes have become more popular and, rather than being seen as an isolated service, should be seen as an excellent companion to the modern data warehouse. Data lakes offer the flexibility to process varied data types at a variety of frequencies, distilling value at every stage, which can then be passed into the modern data warehouse and analyzed by the end users alongside the more traditional measures and stats.

In recent years, many organizations have been struggling with the issues associated with on-premises data warehousing and are now looking to modernize. The rise of the modern data warehouse has already begun, and the goal of this book is to ensure every reader can reap the full benefit.

Getting Started

Microsoft Azure is a comprehensive cloud platform that provides the ability to build Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS) components on both Microsoft-specific services and also third-party and open source technologies. Free trials are available for Microsoft Azure that provide 30-day access and roughly £150/$200 worth of Azure credit. This should allow you to explore most if not all services in this book and gather more of a practical understanding of their implementation. There are also free tiers available for many services that provide sufficient amounts of features for reviewing. Alternatively, you or your company may already have an existing Azure subscription which could then be used to experiment with the technologies listed in this book.

Multi-region Support

A core element of Azure is its multi-region support. As you may know, the cloud is really just someone else's computer, and in this case, the computer belongs to Microsoft and it is stored in a massive data center. It is these data centers that comprise an Azure region. If you are based in America, then you can pick from a range of regions, one of which will be your local region and will likely offer you the lowest latency; you could however deploy resources to a European region if you knew you were supporting customers in that part of the world. Most regions have a paired region which is used for disaster recovery scenarios, but on the whole it is best to keep related resources in the same region. This is to avoid data egress fees which are charged of data that has to be moved out of a region and into another. Note, Azure does not charge data ingress fees.

Resource Groups and Tagging

Once an Azure subscription has been set up, there are a few recommendations to help you organize the subscription. First is the resource group. The resource group is the root container for all single resources and allows a logical grouping for different services that relate to a single system. For example, a modern data warehouse may sit within a resource group that contains an Azure Data Factory, an Azure SQL Database, and an Azure Data Lake Gen 2 (ADL Gen2) account. The resource group means that admins can assign permissions to that single level and control permissions for the entire system. As the subscription gets more use, you should begin creating resource groups per project or application, per environment, so for a single data warehouse, you may have a development, test, and production resource group, each with different permissions.

Another useful technique is to use tags. Tags allow admins to label different resources so that they can be found easily and tracked against different departments, even if they are stored in the same resource group. Common tags include

- Cost center
- Owner
- Creator
- Application

However, many others could be useful to your organization.

Azure Security

From a security standpoint, Azure is an incredibly well-trusted platform. With over 90 compliance certificates in place, including many that are industry or region specific, no cloud platform has a more comprehensive portfolio. Microsoft has invested over one billion US dollars into the security of the Azure platform, having an army of cyber security experts at hand to keep your data safe. These facts and figures offer assurance that the cloud platform is secure; however, within your environment, it is important to properly secure data against malicious employees or external services. This is where service principals are employed. These are service accounts that can be assigned access to many of the resources in the resource group without any human employees having access to the data, ensuring the most sensitive datasets can remain protected.

Modernizing a data platform is no easy task. There are a lot of new terminology and new technologies to understand. In order to work with the demos and walk-throughs in this book, I have prepared some initial resources to review so that there is a common understanding.

Tools of the Trade

There are some tools that will make these technologies easier to use. These are easy to download and work with and in most cases are cross platform compatible, meaning they can work on Apple Macs and Windows machines. The following list explains the key tools that will come in handy throughout this book and what technologies they will assist with:

- **Visual Studio**: 2019 is the current version and is the primary integrated development environment (IDE) when working with Azure and other Microsoft-based technologies.
- **Visual Studio SQL Server Data Tools**: This add-in for Visual Studio gives developers the ability to create database projects and other BI-related projects such as Analysis Services.
- **Microsoft Azure Storage Explorer**: This lightweight tool allows developers to connect to cloud storage accounts and access them as if they were local to their PC. When working with data lakes, this can be very useful.
- **SQL Server Management Studio**: If you are based on a Windows environment, then this is a very powerful tool for monitoring and managing your SQL databases that has been trusted for years.
- **Azure Data Studio**: This is a cross platform version of SQL Server Management Studio. Essentially, this is the go-to place for managing and monitoring any Microsoft SQL environment.

Glossary of Terms

With many new technologies being incorporated into the data platform, a glossary of terms is important to help introduce a conformed understanding. Additionally, many of these terms can be searched online which will allow development teams and architects to research the technologies more fully. The goal of this glossary, shown in Table 1-1, is to act as a point of reference for readers of this book, in case some terminology is new to them.

Table 1-1. *Common Azure Terms*

Term	Definition
Azure Automation	A service that allows for the automated execution of PowerShell scripts in the Azure platform. Scripts can be scheduled or executed using a web hook. Parameters can also be passed in where needed
Azure Synapse Analytics	A massively parallel processing (MPP) engine used for storing and processing large structured datasets in Azure using the SQL server engine over a distributed cluster of computers
Azure SQL Database	A symmetric processing engine that specializes in OLTP workloads in Azure. Equivalent to a single database in an on-premises SQL server environment
Azure Data Factory	A cloud-based integration engine capable of copying and transforming data at scale
Azure Blob Storage	A highly scalable storage platform that can hold data of all types and sizes
Azure Data Lake Gen 2	Built on Azure Blob Storage with the addition of hierarchical namespaces to allow for granular security with AAD integration
Azure Key Vault	A REST-based cloud secret manager that is tightly integrated into the Azure platform
Azure Cosmos DB	A highly scalable document database that uses a variety of APIs to implement different storage paradigms such as SQL, Graph, No SQL, and key value pair
Azure Databricks	A PaaS implementation of Spark, allowing you to scale and pause your cluster with a rich notebook environment
Microsoft Power BI	A market leading data visualization and end-to-end BI tool offering excellent data exploration and collaboration capabilities
Analysis Services	A semantic layer offering from Microsoft that uses Fact and Dimension tables to create a compressed and optimized data model

Naming Conventions

All development projects can benefit from a rigorous naming convention in my opinion and so a modern data warehouse is no different. A good naming convention should supply those that read the name enough detail to understand what the object is and roughly what it does. Additionally, a naming convention clears up any debate about what a particular thing should be called, as the formula to produce the name already exists. The naming convention included here is the standard recommended by Azure, which I have simply described in a shorter format.

The name of a resource is broken down into several pieces, and so the following list describes each section of the name. In the following, I will offer some examples of resource names, assuming the project for the book is called "Modern Data Warehouse in Azure":

- **Department, business unit or project**: This could be "mrkt" for marketing, "fin" for finance, or "sls" for sales.
- **Application or service name**: For example, a SQL database would be "sqldb," a Synapse Analytics database would be "syndb," an Azure Data Factory would be "adf."
- **Environment**: This could be "dev," "test," "sit," "prod," to name a few.
- **Deployment region**: This is the region in which the resource is located and is usually abbreviated such that East US would become "eus" and North Europe would become "neu."

In Table 1-2, I have given examples of some common data warehousing resources alongside their suggested names.

Table 1-2. *Example Azure resource names*

Resource	Resource Name
Azure SQL Database	mdwa-sqldb-dev-eus
Azure Synapse Analytics	mdwa-syndb-dev-eus
Azure Data Factory	mdwa-adf-dev-eus
Azure Data Lake Gen 2	mdwaadlsdeveus
Azure Key Vault	mdwa-kv-dev-eus

CHAPTER 2

The SQL Engine

The focus of this chapter is to break open the mysteries of each SQL storage engine and understand why a particular flavor of Azure SQL technology suits one scenario over another. We will analyze the underlying architecture of each service so that development choices can be well informed and well reasoned. Once we understand how each implementation of the SQL engine in Azure processes and stores data, we can look at the direction Microsoft is taking that technology and forecast whether the same choice would be made in the future. The knowledge gained in this chapter should provide you with the capability to understand your source data and therefore to choose which SQL engine should be used to store and process that data.

Later in this book, we will move out of the structured SQL world and discuss how we can utilize Azure data lake technology to more efficiently work with our data; however, those services are agnostic to the SQL engine that we decide best suits our use case and therefore can be decided upon later. As a primary focus, we must understand our SQL options, and from there, we can tailor our metadata, preparation routines, and development tools to suit that engine.

The Four Vs

The Microsoft Azure platform has a wealth of data storage options at the user's disposal, each with different features and traits that make them well suited for a given type of data and scenario. Given the flexible and dynamic nature of cloud computing, Microsoft has built a comprehensive platform that ensures all varieties of data can be catered for. The acknowledgment of the need to cater to differing types of data gets neatly distilled into what is known in the data engineering world as "The 3 Vs" – volume, variety, and velocity.

Any combination of volume, variety, and velocity can be solved using a storage solution in the Azure platform. Often people refer to a fourth V being "value" which I think is a worthy addition as the value can often get lost in the volume.

M. How, *The Modern Data Warehouse in Azure*, https://doi.org/10.1007/978-1-4842-5823-1_2

As the volume increases, the curation process to distil value from data becomes more complex, and therefore, specific tools and solutions can be used to help that process, validating the need for a fourth V. When attempting to tackle any one or combination of the four Vs, it is important to understand the full set of options available so that a well-informed decision can be made. Understanding the reasons why a certain technology should be chosen over another is essential to any development process, as this can then inform the code, structure, and integration of that technology.

To use an example, if you needed to store a large amount of enterprise data that was a complete mix of file types and sizes, you would use an Azure Storage account. This would allow you to organize your data into a clear structure and efficiently increase your account size as and when you need. The aspects of that technology help to reduce the complexities of dealing with large-scale data and remove any barriers to entry. Volume, check. Variety, check.

Alternatively, if the requirement was to store JavaScript Object Notation (JSON) documents so that they can be efficiently queried, then the best option would be to utilize Cosmos DB. While there is nothing stopping JSON data being stored in Blob Storage, the ability to index and query JSON data using Cosmos DB make this an obvious choice. The guaranteed latency and throughput options of Cosmos DB mean that high-velocity data is easily ingested. When the volume begins to increase, then Cosmos DB will scale with it. Velocity, check. Volume, check.

Moving to a data warehouse, we know we will have a large amount of well-structured, strongly typed data that needs to rapidly serve up analytical insight. We need a SQL engine. Crucially, this is where the fourth V, "value," comes into play. Datasets being used to feed a data warehouse may contain many attributes that are not especially valuable, and good practice dictates that these attributes are trimmed off before arriving in the data warehouse. The golden rule is that data stored in a data warehouse should be well curated and of utmost value. A SQL engine makes surfacing that valuable data easy, and further to that, no other storage option can facilitate joining of datasets to produce previously uncovered value as effortlessly as a SQL engine can. Value, check.

However, a wrinkle in the decision process is that Azure provides two types of SQL engine to choose from; each can tackle any challenge in the four Vs; however, it is wise to understand which engine solves which "V" best. Understanding the nuances of each flavor of Azure SQL will help developers make informed decisions about how to load, query, and manage the data warehouse.

The first SQL engine we will examine in this chapter is Azure Synapse Analytics (formerly Azure SQL Data Warehouse). This massively parallel processing (MPP) service provides scalability, elasticity, and concurrency, all underpinned by the well-loved Microsoft SQL server engine. The clue is certainly in the former title; this is a good option for data warehousing. However, there are other factors that mean this may not be the right choice in all scenarios. While Azure Synapse Analytics has a wealth of optimizations targeted at data warehousing, there are some reasons why the second SQL option, Azure SQL Database, may be more suitable.

Azure SQL Database is an OLTP type system that is optimized for reads and writes; however, it has some interesting features that make it a great candidate for a data warehouse environment. The recent advent of Azure SQL Database Hyperscale means that Azure SQL Database can scale up to 100 TB and provide additional read-only compute nodes to serve up analytical data. A further advantage is that Azure SQL Database has intelligent query processing and can be highly reactive to changes in runtime conditions allowing for peak performance to be maintained at critical times. Finally, there are multiple deployment options for Azure SQL Database that include managed instances and elastic pools. In essence, a managed instance is a full-blown SQL server instance deployed to the cloud and provides the closest match to an existing on-premises Microsoft SQL server implementation in Azure. Elastic pool databases utilize a single pool of compute resource to allow for a lower total cost of ownership as databases can consume more and less resources from the pool rather than having to be scaled independently.

Azure Synapse Analytics

When implementing an on-premises data warehouse, there are many constraints placed upon the developer. Initially there is the hassle of setting up and configuring the server, and even if this is taken care of already, there is always a maintenance and management overhead that cannot be ignored. Once the server is set up, further thought needs to be applied to file management and growth. In addition, the data warehouse itself is limited to the confines of the physical box, and often large databases have to utilize complex storage solutions to mitigate this issue.

However, if you are reading this book, then it is clear you are no longer interested in this archaic and cumbersome approach to data warehousing. By making the move up to the Azure cloud, you can put the days of server management behind you, safe in the knowledge that Microsoft will take care of all that. And what's more, Azure does not just provide a normal SQL instance that is purely serverless; they have restructured the underlying architecture entirely so that it is tailored for the cloud environment. This is then extended further to the point that Azure Synapse Analytics is not only purpose-built for the cloud but purpose-built for large-scale data warehousing.

Understanding Distributions

A key factor that needs to be understood when working with Azure Synapse Analytics is that of distributions. In a standard SQL server implementation, you are working in a symmetric multi-processing (SMP) environment which means there is a single storage point coupled to a set of CPUs and queries are parallelized across those CPUs using a service bus. The main problem here is that all the CPUs need to access the same storage and this can become a bottleneck, especially when running large analytical queries.

When you begin using Azure Synapse Analytics, you are now in a massively parallel processing (MPP) environment.

There are a number of key differences between SMP and MPP environments, and they are illustrated in Figure 2-1. The most important is that storage is now widely distributed and coupled to a specific amount of compute. The benefit here is that each node of the engine is essentially a separate SQL database and can access its own storage separately from all the other nodes without causing contention.

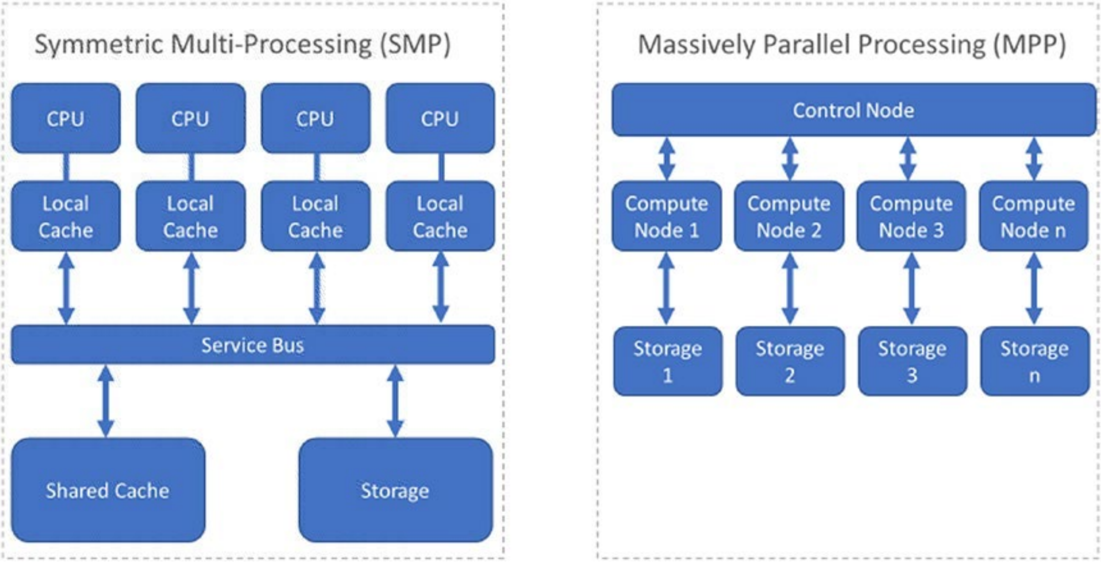


Figure 2-1. *Diagram of SMP vs. MPP*