LEARNING MADE EASY

Data Lakes

for dummies®

A Wiley Brand

Turn data sources into usable analytics

Refine and enrich your raw data

Choose the right vendor to add quality to data

Alan Simon

Data Lakes

for dummies®
A Wiley Brand

Data Lakes

by Alan Simon

for dummies®
A Wiley Brand

Data Lakes For Dummies®

Published by: **John Wiley & Sons, Inc.,** 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Published simultaneously in Canada

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit https://hub.wiley.com/community/support/dummies.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2021939570

ISBN 978-1-119-78616-0 (pbk); ISBN 978-1-119-78617-7 (ebk); ISBN 978-1-119-78618-4 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents at a Glance

Table of Contents

Introduction

In December 1995, I wrote an article for *Database Programming & Design* magazine entitled "I Want a Data Warehouse, So What Is It Again?" A few months later, I began writing *Data Warehousing For Dummies* (Wiley), building on the article's content to help readers make sense of first-generation data warehousing.

Fast-forward a quarter of a century, and I could very easily write an article entitled "I Want a Data Lake, So What Is It Again?" This time, I'm cutting right to the chase with *Data Lakes For Dummies.* To quote a famous former baseball player named Yogi Berra, it's déjà vu all over again!

Nearly every large and upper-midsize company and governmental agency is building a data lake or at least has an initiative on the drawing board. That's the good news.

The not-so-good news, though, is that you'll find a disturbing lack of agreement about data lake architecture, best practices for data lake development, data lake internal data flows, even what a data lake actually *is!* In fact, many first-generation data lakes have fallen short of original expectations and need to be rearchitected and rebuilt.

As with data warehousing in the mid-'90s, the data lake concept today is still a relatively new one. Consequently, almost everything about data lakes — from its very definition to alternatives for integration with or migration from existing data warehouses — is still very much a moving target. Software product vendors, cloud service providers, consulting firms, industry analysts, and academics often have varying — and sometimes conflicting — perspectives on data lakes. So, how do you navigate your way across a data lake when the waters are especially choppy and you're being tossed from side to side?

That's where *Data Lakes For Dummies* comes in.

About This Book

Data Lakes For Dummies helps you make sense of the ABCs — acronym anarchy, buzzword bingo, and consulting confusion — of today's and tomorrow's data lakes.

This book is not only a tutorial about data lakes; it also serves as a reference that you may find yourself consulting on a regular basis. So, you don't need to memorize large blocks of content (there's no final exam!) because you can always go back to take a second or third or fourth look at any particular point during your own data lake efforts.

Right from the start, you find out what your organization should expect from all the time, effort, and money you'll put into your data lake initiative, as well as see what challenges are lurking. You'll dig deep into data lake architecture and leading cloud platforms and get your arms around the big picture of how all the pieces fit together.

One of the disadvantages of being an early adopter of any new technology is that you sometimes make mistakes or at least have a few false starts. Plenty of early data lake efforts have turned into more of a data dump, with tons of data that just isn't very accessible or well organized. If you find yourself in this situation, fear not: You'll see how to turn that data dump into the data lake you originally envisioned.

I don't use many special conventions in this book, but you should be aware that sidebars (the gray boxes you see throughout the book) and anything marked with the Technical Stuff icon are all skippable. So, if you're short on time, you can pass over these pieces without losing anything essential. On the other hand, if you have the time, you're sure to find fascinating information here!

Within this book, you may note that some web addresses break across two lines of text. If you're reading this book in print and want to visit one of these web pages, simply key in the web address exactly as it's noted in the text, pretending as though the line break doesn't exist. If you're reading this as an e-book, you've got it easy — just click the web address to be taken directly to the web page.

Foolish Assumptions

The most relevant assumption I've made is that if you're reading this book, you either are or will soon be working on a data lake initiative.

Maybe you're a data strategist and architect, and what's most important to you is sifting through mountains of sometimes conflicting — and often incomplete — information about data lakes. Your organization already makes use of earlier-generation data warehouses and data marts, and now it's time to take that all-important next step to a data lake. If that's the case, you're definitely in the right place.

If you're a developer or data architect who is working on a small subset of the overall data lake, your primary focus is how a particular software package or service works. Still, you're curious about where your daily work fits into your organization's overall data lake efforts. That's where this book comes in: to provide context and that "aha!" factor to the big picture that surrounds your day-to-day tasks.

Or maybe you're on the business and operational side of a company or governmental agency, working side by side with the technology team as they work to build an enterprise-scale data environment that will finally support the entire spectrum of your organization's analytical needs. You don't necessarily need to know too much about the techie side of data lakes, but you absolutely care about building an environment that meets today's and tomorrow's needs for data-driven insights.

The common thread is that data lakes are part of your organization's present and future, and you're seeking an unvarnished, hype-free, grounded-in-reality view of data lakes today and where they're headed.

In any event, you don't need to be a technical whiz with databases, programming languages such as Python, or specific cloud platforms such as Amazon Web Services (AWS) or Microsoft Azure. I cover many different technical topics in this book, but you'll find clear explanations and diagrams that don't presume any prerequisite knowledge on your part.

Icons Used in This Book

As you read this book, you encounter icons in the margins that indicate material of particular interest. Here's what the icons mean:

TIP

These are the tricks of the data lake trade. You can save yourself a great deal of time and avoid more than a few false starts by following specific tips collected from the best practices (and learned from painful experiences) of those who preceded you on the path to the data lake.

WARNING

Data lakes are often filled with dangerous icebergs. (Okay, bad analogy, but you hopefully get the idea.) When you're working on your organization's data lake efforts, pay particular attention to situations that are called out with this icon.

TECHNICAL STUFF

If you're more interested in the conceptual and architectural aspects of data lakes than the nitty-gritty implementation details, you can skim or even skip material that is accompanied by this icon.

REMEMBER

Some points are so critically important that you'll be well served by committing them to memory. You'll even see some of these points repeated later in the book because they tie in with other material. This icon calls out this crucial content.

Beyond the Book

In addition to the material in the print or e-book you're reading right now, this product comes with a free Cheat Sheet for the three types of data for your data lake, four zones inside your data lake, five phases to building your data lake, and more. To access the Cheat Sheet, go to `www.dummies.com` and type **Data Lakes For Dummies Cheat Sheet** in the Search box.

Where to Go from Here

Now it's time to head off to the lake — the data lake, that is! If you're totally new to the subject, you don't want to skip the chapters in Part 1 because they'll provide the foundation for the rest of the book. If you already have some exposure to data lakes, I still recommend that you at least skim Part 1 to get a sense of how to get beyond all the hype, buzzwords, and generalities related to data lakes.

You can then read the book sequentially from front to back or jump around as needed. Whatever path works best for you is the one you should take.

1
Getting Started with Data Lakes

IN THIS PART . . .

Separate the data lake reality from the hype.

Steer your data lake efforts in the right direction.

Diagnose and avoid common pitfalls that can dry up your data lake.

IN THIS CHAPTER

- **Defining and scoping the data lake**
- **Diving underwater in the data lake**
- **Dividing up the data lake**
- **Making sense of conflicting terminology**

Chapter 1
Jumping into the Data Lake

The lake is the place to be this season — the data lake, that is!

Just like the newest and hottest vacation destination, everyone is booking reservations for a trip to the data lake. Unlike a vacation, though, you won't just be spending a long weekend or a week or even the entire summer at the data lake. If you and your work colleagues do a good job, your data lake will be your go-to place for a whole *decade* or even longer.

What Is a Data Lake?

Ask a friend this question: "What's a lake?" Your friend thinks for a moment, and then gives you this answer: "Well, it's a big hole in the ground that's filled with water."

Technically, your friend is correct, but that answer also is far from detailed enough to really tell you what a lake actually is. You need more specifics, such as:

- How big, dimension-wise (how long and how wide)
- How deep that "big hole in the ground" goes
- How much variability there is from one lake to another in terms of those length, width, and depth dimensions (the Great Lakes, anyone?)
- How much water you'll find in the lake and how much that amount of water may vary among different lakes
- Whether a lake contains freshwater or saltwater

Some follow-up questions may pop into your mind as well:

- A pond is also a big hole in the ground that's filled with water, so is a lake the same as a pond?
- What distinguishes a lake from an ocean or a sea?
- Can a lake be physically connected to another lake?
- Can the dividing line between two states or two countries be in the middle of a lake?
- If a lake is empty, is it still considered a lake?
- If one lake leaves Chicago, heading east and travels at 100 miles per hour, and another lake heads west from New York . . . oh wait, wrong kind of word problem, never mind. . . .

So many missing pieces of the puzzle, all arising from one simple question!

You'll find the exact same situation if you ask someone this question: "What's a data lake?" In fact, go ahead and ask your favorite search engine that question. You'll find dozens of high-level definitions that will almost certainly spur plenty of follow-up questions as you try to get your arms around the idea of a data lake.

TIP

Here's a better idea: Instead of filtering through all that varying — and even conflicting — terminology and then trying to consolidate all of it into a single comprehensive definition, just think of a data lake as the following:

> A solidly architected, logically centralized, highly scalable environment filled with different types of analytic data that are sourced from both inside and outside your enterprise with varying latency, and which will be the primary go-to destination for your organization's data-driven insights

Wow, that’s a mouthful! No worries: Just as if you were eating a gourmet fireside meal while camping at your favorite lake, you can break up that definition into bite-size pieces.

Rock-solid water

A data lake should remain viable and useful for a long time after it becomes operational. Also, you’ll be continually expanding and enhancing your data lake with new types and forms of data, new underlying technologies, and support for new analytical uses.

REMEMBER

Building a data lake is more than just loading massive amounts of data into some storage location.

To support this near-constant expansion and growth, you need to ensure that your data lake is well architected and solidly engineered, which means that the data lake

- Enforces standards and best practices for data ingestion, data storage, data transmission, and interchange among its components and data delivery to end users
- Minimizes workarounds and temporary interfaces that have a tendency to stick around longer than planned and weaken your overall environment
- Continues to meet your predetermined metrics and thresholds for overall technical performance, such as data loading and interchange, as well as user response time

Think about a resort that builds docks, a couple of lakeside restaurants, and other structures at various locations alongside a large lake. You wouldn’t just hand out lumber, hammers, and nails to a bunch of visitors and tell them to start building without detailed blueprints and engineering diagrams. The same is true with a data lake. From the first piece of data that arrives, you need as solid a foundation as possible to help keep your data lake viable for a long time.

A really great lake

You’ll come across definitions and descriptions that tell you a data lake is a centralized store of data, but that definition is only partially correct.

A data lake is *logically* centralized. You can certainly think of a data lake as a single place for your data, instead of having your data scattered among different databases. But in reality, even though your data lake is logically centralized, its data is *physically* decentralized and distributed among many different underlying servers.

TECHNICAL STUFF

The data services that you use for your data lake, such as the Amazon Simple Storage Service (S3), the Microsoft Azure Data Lake Storage (ADLS), or the Hadoop Distributed File System (HDFS) manage the distribution of data among potentially numerous servers where your data is actually stored. These services hide the physical distribution from almost everyone other than those who need to manage the data at the server storage level. Instead, they present the data as being logically part of a single data lake. Figure 1-1 illustrates how logical centralization accompanies physical decentralization.

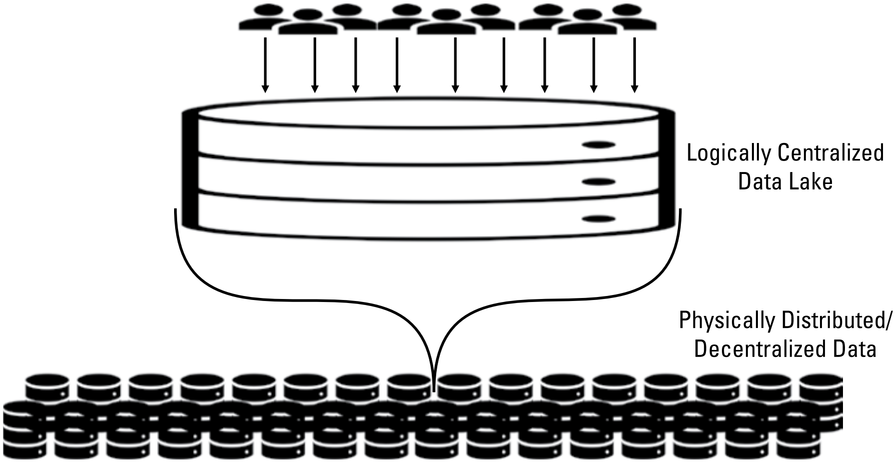

FIGURE 1-1: A logically centralized data lake with underlying physical decentralization.

Expanding the data lake

How big can your data lake get? To quote the old saying (and to answer a question with a question), how many angels can dance on the head of a pin?

Scalability is best thought of as "the ability to expand capacity, workload, and missions without having to go back to the drawing board and start all over." Your data lake will almost always be a cloud-based solution (see Figure 1-2). Cloud-based platforms give you, in theory, infinite scalability for your data lake. New servers and storage devices (discs, solid state devices, and so on) can be incorporated into your data lake on demand, and the software services manage and control these new resources along with those that you're already using. Your data lake contents can then expand from hundreds of terabytes to petabytes, and then to exabytes, and then zettabytes, and even into the ginormousbyte range. (Just kidding about that last one.)

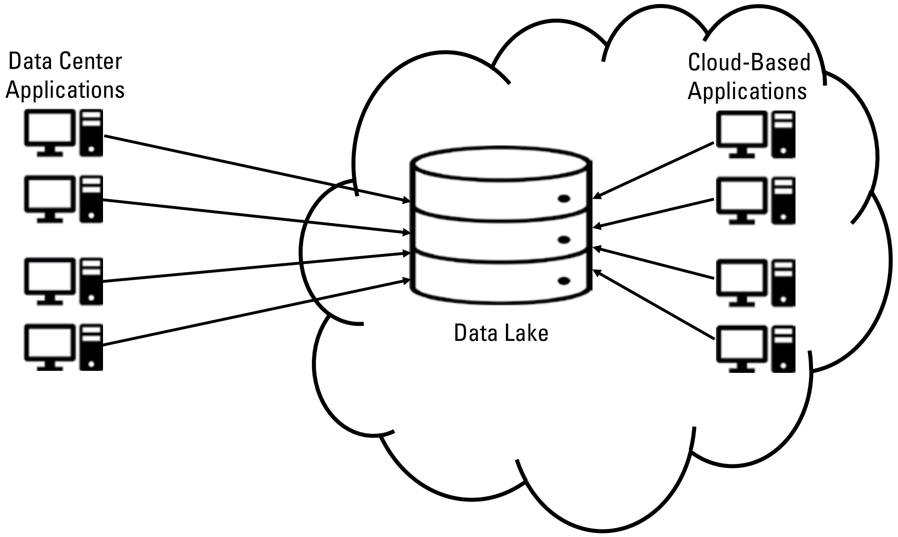

FIGURE 1-2: Cloud-based data lake solutions.

TIP

Cloud providers give you pricing for data storage and access that increases as your needs grow or decreases if you cut back on your functionality. Basically, your data lake will be priced on a pay-as-you-go basis.

Some of the very first data lakes that were built in the Hadoop environment may reside in your corporate data center and be categorized as *on-prem* (short for *on-premises,* meaning "on your premises") solutions. But most of today's data lakes are built in the Amazon Web Services (AWS) or Microsoft Azure cloud environments. Given the ever-increasing popularity of cloud computing, it's highly unlikely that this trend of cloud-based data lakes will reverse for a long time, if ever.

As long as Amazon, Microsoft, and other cloud platform providers can keep expanding their existing data centers and building new ones, as well as enhancing the capabilities of their data management services, then your data lake should be able to avoid scalability issues.

TECHNICAL STUFF

A multiple-component data lake architecture (see Chapter 4) further helps overcome performance and capacity constraints as your data lake grows in size and complexity, providing even greater scalability.

More than just the water

Think of a data lake as being closer to a lake resort rather than just the lake — the body of water — in its natural state. If you were a real estate developer, you might buy the property that includes the lake itself, along with plenty of acreage surrounding the lake. You'd then develop the overall property by building cabins, restaurants, boat docks, and other facilities. The lake might be the centerpiece of the overall resort, but its value is dramatically enhanced by all the additional assets that you've built surrounding the lake.

REMEMBER

A data lake is an entire environment, not just a gigantic collection of data that is stored within a data service such as Amazon S3 or Microsoft ADLS.

In addition to data storage, a data lake also includes the following:

- One or (usually) more mechanisms to move data from one part of the data lake to another.
- A catalog or directory that helps keep track of what data is where, as well as the associated rules that apply to different groups of data; this is known as *metadata.*
- Capabilities that help unify meanings and business rules for key data subjects that may come into the data lake from different applications and systems; this is known as *master data management.*
- Monitoring services to track data quality and accuracy, response time when users access data, billing services to charge different organizations for their usage of the data lake, and plenty more.

Different types of data

If your data lake had a motto, it might be "All data are created equal."

In a data lake, data is data is data. In other words, you don't need to make special accommodations for more complex types of data than you would for simpler forms of data.

Your data lake will contain structured data, unstructured data, and semi-structured data (see Figure 1-3). The following sections cover these types of data in more detail.

Structured data: Staying in your own lane

You're probably most familiar with *structured data,* which is made up of numbers, shorter-length character strings, and dates. Traditionally, most of the applications you've worked with have been based on structured data. Structured data is commonly stored in a relational database such as Microsoft SQL Server, MySQL, or Oracle Database.

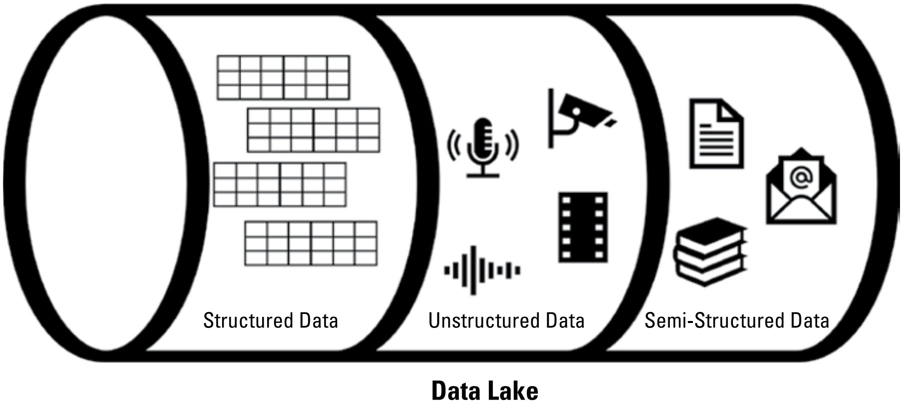

FIGURE 1-3: Different types of data in your data lake.

In a database, you define columns (basically, fields) for each of your pieces of structured data, and each column is rigidly and precisely defined with the following:

- **A data type,** such as INTEGER, DECIMAL, CHARACTER, DATE, DATETIME, or something similar
- **The size of the field,** either explicitly declared (for example, how many characters a CHARACTER column will contain) or implicitly declared (the system-defined maximum number for an INTEGER or how a DATE column is structured)
- **Any specific rules that apply to a data column or field,** such as the permissible range of values (for example, a customer's age must be between 18 and 130) or a list of allowable values (for example, an employee's current status can only be FULL-TIME, PART-TIME, TERMINATED, or RETIRED)
- **Any additional constraints,** such as primary and foreign key designations, or *referential integrity* (rules that specify consistency for certain columns across multiple database tables)

Unstructured data: A picture may be worth ten million words

Unstructured data is, by definition, data that lacks a formally defined structure. Images (such as JPEGs), audio (such as MP3s), and videos (such as MP4s or MOVs) are common forms of unstructured data.

Semi-structured data: Stuck in the middle of the lake

Semi-structured data sort of falls in between structured and unstructured data. Examples include a blog post, a social media post, text messages, an email message, or a message from Slack or Microsoft Teams. Leaving aside any embedded or attached images or videos for a moment, all these examples consist of a long string of letters, numbers, and special characters. However, there's no particular structure assigned to most of these text strings other than perhaps a couple of lines of heading information. The body of an email may be very short — only a line or two — while another email can go on for many long paragraphs.

In your data lake, you need to have all these types of data sitting side by side. Why? Because you'll be running analytics against the data lake that may need more than one form of data. For example, you receive and then analyze a detailed report of sales by department in a large department store during the past month.

Then, after noticing a few anomalies in the sales numbers, you pull up in-store surveillance video to analyze traffic versus sales to better understand how many customers may be looking at merchandise but deciding not to make a purchase. You can even combine structured data from scanners with your unstructured video data as part of your analysis.

If you had to go to different data storage environments for your sales results (structured data) and then the video surveillance (unstructured data), your overall analysis is dramatically slowed down, especially if you need to integrate and cross-reference different types of data. With a data lake, all this data is sitting side by side, ready to be delivered for analysis and decision-making.

TECHNICAL STUFF

In their earliest days, relational databases only stored structured data. Later, they were extended with capabilities to store structured and unstructured data. Binary large objects (BLOBs) were a common way to store images and even video in a relational database. However, even an *object-extended* relational database doesn't make a good platform for a data lake when compared with modern data services such as Amazon S3 or Microsoft ADLS.

Different water, different data

A common misconception is that you store "all your data" in your data lake. Actually, you store all or most of your *analytic* data in a data lake. Analytic data is, as you may suspect from the name, data that you're using for analytics. In contrast, you use *operational* data to run your business.