O'REILLY®

Data Quality Fundamentals

A Practitioner's Guide to Building Trustworthy Data Pipelines

Sponsored by

MC MONTE CARLO

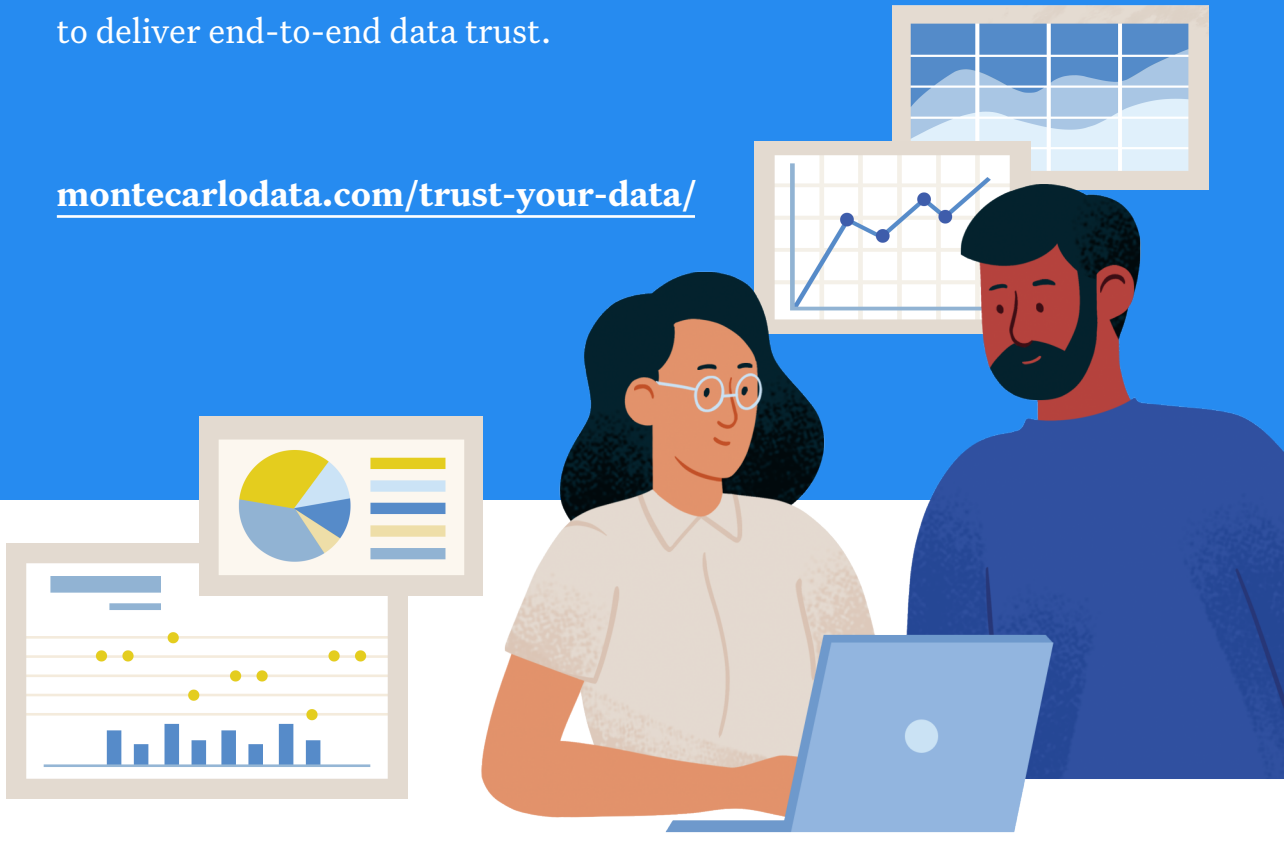
Barr Moses,
Lior Gavish &
Molly Vorwerck

MC MONTE CARLO

Trust your data.

Learn how modern data teams rely on
Monte Carlo's Data Observability Platform
to deliver end-to-end data trust.

montecarlodata.com/trust-your-data/



Praise for *Data Quality Fundamentals*

Data engineers, ETL programmers, and entire data pipeline teams need a reference and testing guide like this! As I did, they will learn the building blocks, processes, and tooling that help ensure the quality of data-intensive applications.

This book adds fresh perspectives and practical test scenarios that expand the wisdom to test modern data pipelines.

—Wayne Yaddow, Data and ETL Quality Analyst

Your data investments, infrastructure, and insights don't matter at all if you can't trust your data. Barr, Lior, and Molly have done a tremendous job in breaking down the fundamentals of what trusting your data means and have created a very practical framework to implement data quality in enterprises. A must-read for anyone who cares about data quality.

—Debashis Saha, Data Leader
AppZen, Intuit, and eBay

As data architecture becomes increasingly distributed and the accountability for data increasingly decentralized, the focus on data quality will continue to grow. *Data Quality Fundamentals* provides an important resource for engineering teams that are serious about improving the accuracy, reliability, and trust of their data through some of today's most significant technologies and processes.

—Mammad Zadeh, Data Leader and
Former VP of Engineering at Intuit

Data Quality Fundamentals

A Practitioner's Guide to Building Trustworthy Data Pipelines

Barr Moses, Lior Gavish, and Molly Vorwerck

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

To Rae and Robert, who keep things in perspective, no matter where we look.

To the Monte Carlo jellyfish and the data reliability pioneers—you know who you are.
So grateful to be on this journey with you.

Table of Contents

Preface

If you've experienced any of the following scenarios, raise your hand (or, you can just nod in solidarity—there's no way we'll know otherwise):

- Five thousand rows in a critical (and relatively predictable) table suddenly turns into five hundred, with no rhyme or reason.
- A broken dashboard causes an executive dashboard to spit null values.
- A hidden schema change breaks a downstream pipeline.
- And the list goes on.

This book is for everyone who has suffered from unreliable data, silently or with muffled screams, and wants to do something about it. We expect that these individuals will come from data engineers, data analytics, or data science backgrounds, and be actively involved in building, scaling, and managing their company's data pipelines.

On the surface, it may seem like *Data Quality Fundamentals* is a manual about how to clean, wrangle, and generally make sense of data—and it is. But more so, this book tackles best practices, technologies, and processes around building more reliable data systems and, in the process, cultivating data trust with your team and stakeholders.

In Chapter 1, we'll discuss why data quality deserves attention now, and how architectural and technological trends are contributing to an overall decrease in governance and reliability. We'll introduce the concept of "data downtime," and explain how it harkens back to the early days of site reliability engineering (SRE) teams and how these same DevOps principles can apply to your data engineering workflows as well.

In Chapter 2, we'll highlight how to build more resilient data systems by walking through how you can solve for and measure data quality across several key data pipeline technologies, including data warehouses, data lakes, and data catalogs. These three foundational technologies store, process, and track data health preproduction, which naturally leads us into Chapter 3, where we'll walk through how to collect, clean, transform, and test your data with quality and reliability in mind.

Next, Chapter 4 will walk through one of the most important aspects of the data reliability workflow—proactive anomaly detection and monitoring—by sharing how to build a data quality monitor using a publicly available data set about exoplanets. This tutorial will give readers the opportunity to directly apply the lessons they've learned in *Data Quality Fundamentals* to their work in the field, albeit at a limited scale.

Chapter 5 will provide readers with a bird's-eye view into what it takes to put these critical technologies together and architect robust systems and processes that ensure data quality is measured and maintained no matter the use case. We'll also share how best-in-class data teams at Airbnb, Uber, Intuit, and other companies integrate data reliability into their day-to-day workflows, including setting SLAs, SLIs, and SLOs, and building data platforms that optimize for data quality across five key pillars: freshness, volume, distribution, schema, and lineage.

In Chapter 6, we'll dive into the steps necessary to actually react to and fix data quality issues in production environments, including data incident management, root cause analysis, postmortems, and establishing incident communication best practices. Then, in Chapter 7, readers will take their understanding of root cause analysis one step further by learning how to build field-level lineage using popular and widely adopted open source tools that should be in every data engineer's arsenal.

In Chapter 8, we'll discuss some of the cultural and organizational barriers data teams must cross when evangelizing and democratizing data quality at scale, including best-in-class principles like treating your data like a product, understanding your company's RACI matrix for data quality, and how to structure your data team for maximum business impact.

In Chapter 9, we'll share several real-world case studies and conversations with leading minds in the data engineering space, including Zhamak Dehghani, creator of the data mesh, António Fitas, whose team bravely shares their story of how they're migrating toward a decentralized (and data quality first!) data architecture, and Alex Tverdohleb, VP of Data Services at Fox and a pioneer of the "controlled freedom" data management technique. This patchwork of theory and on-the-ground examples will help you visualize how several of the technical and process-driven data quality concepts we highlight in Chapters 1 through 8 can come to life in stunning color.

And finally, in Chapter 10, we finish our book with a tangible calculation for measuring the financial impact of poor data on your business, in human hours, as a way to help readers (many of whom are tasked with fixing data downtime) make the case with leadership to invest in more tools and processes to solve these problems. We'll also highlight four of our predictions for the future of data quality as it relates to broader industry trends, such as distributed data management and the rise of the data lakehouse.

At the very least, we hope that you walk away from this book with a few tricks up your sleeve when it comes to making the case for prioritizing data quality and reliability across your organization. As any seasoned data leader will tell you, data trust is never built in a day, but with the right approach, incremental progress can be made—pipeline by pipeline.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

This element signifies a tip or suggestion.

This element signifies a general note.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at *https://oreil.ly/data-quality-fundamentals-code*.

If you have a technical question or a problem using the code examples, please send an email to *bookquestions@oreilly.com*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant

amount of example code from this book into your product's documentation does require permission.

We appreciate, but generally do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Data Quality Fundamentals* by Barr Moses, Lior Gavish, and Molly Vorwerck (O'Reilly). Copyright 2022 Monte Carlo Data, Inc., 978-1-098-11204-2."

If you feel your use of code examples falls outside fair use or the permission described herein, feel free to contact us at *permissions@oreilly.com*.

O'Reilly Online Learning

For more than 40 years, *O'Reilly Media* has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, visit *https://oreilly.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *https://oreil.ly/data-quality-fundamentals*.

Email *bookquestions@oreilly.com* to comment or ask technical questions about this book.

For news and information about our books and courses, visit *https://oreilly.com*.

Find us on LinkedIn: *https://linkedin.com/company/oreilly-media*.

Follow us on Twitter: *https://twitter.com/oreillymedia*.

Watch us on YouTube: *https://www.youtube.com/oreillymedia*.

Acknowledgments

This book was a labor of love, and for that reason, we have many people to thank.

First, we'd like to thank Jess Haberman, our fearless acquisitions editor, who believed in us every step of the way. When Jess came to us with the idea for a book on data quality, we were taken aback—in the best way possible. We had no idea that a topic—data reliability—that's so near and dear to our hearts would find life outside of our personal blog articles. With her dedication and encouragement, we were able to draft a proposal that set itself apart from what was already published in the space and ultimately write a book that would bring value to other data practitioners struggling with data downtime.

We must also thank Jill Leonard, our development editor, who has served as our Yoda of the entire writing process. From providing invaluable guidance on flow and copy, to being available for pep talks and brainstorming sessions ("Should this chapter go here? What about there? What even *is* a preface?"), Jill was the Jedi who saw us through to the finish line. Our mutual love of cats only helped seal the bond.

We are forever indebted to our technical reviewers, Tristan Baker, Debashis Saha, Wayne Yaddow, Scott Haines, Sam Bail, Joy Payton, and Robert Ansel, for their sharp edits and valuable feedback on multiple drafts of the book. Their passion for bringing DevOps best practices and good data hygiene to the field is an inspiration, and we've been grateful to work with them.

We'd like to acknowledge—and thank a million times over—Ryan Kearns, a contributor to this book whose name could have been on the byline. From spearheading several chapters to offering critical insights on the technologies and processes discussed, this book would not have come together without his assistance. We learn from him every day and are so lucky to call him a dear colleague. In the coming years, Ryan will undoubtedly become one of the most important voices in data engineering and data science.

There were several industry experts and trailblazers we interviewed for this book and various other projects we've pursued over the past year. In no particular order, we'd like to thank Brandon Beidel, Alex Tverdohleb, António Fitas, Gopi Krishnamurthy, Manu Raj, Zhamak Dehghani, Mammad Zadeh, Greg Waldman, Wendy Turner Williams, Zosia Kossowski, Erik Bernhardsson, Jessica Cherny, Josh Wills, Kyle Shannon, Atul Gupte, Chad Sanderson, Patricia Ho, Michael Celentano, Prateek Chawla, Cindi Howson, Debashis Saha, Melody Chien, Ankush Jain, Maxime Beauchemin, DJ Patil, Bob Muglia, Mauricio de Diana, Shane Murray, Francisco Alberini, Mei Tao, Xuanzi Han, and Helena Munoz.

We'd also like to thank Brandon Gubitosa, Sara Gates, and Michael Segner for their assistance with outlines and drafts—and for always encouraging us to "kill our darlings."

We're indebted to our parents, Elisha and Kadia Moses, Motti and Vira Gavish, and Gregg and Barbara Vorwerck, for encouraging us to pursue our passions for data engineering and data quality, from launching a company and category dedicated to the concept, to writing this book. We'd also like to thank Rae Barr Gavish (RBG) for being our number one fan, and Robert Ansel for being our resident SRE, WordPress consultant, and DevOps guru.

And we're forever indebted to our customers, who are helping us pioneer the data observability category and through the process laying the foundations for the future of reliable data at scale.

CHAPTER 1

Why Data Quality Deserves Attention—Now

Raise your hand (or spit out your coffee, sigh deeply, and shake your head) if this scenario rings a bell.

Data is a priority for your CEO, as it often is for digital-first companies, and she is fluent in the latest and greatest business intelligence tools. Your CTO is excited about migrating to the cloud, and constantly sends your team articles highlighting performance measurements against some of the latest technologies. Your downstream data consumers including product analysts, marketing leaders, and sales teams rely on data-driven tools like customer relationship management/customer experience platforms (CRMs/CXPs), content management systems (CMSs), and any other acronym under the sun to do their jobs quickly and effectively.

As the data analyst or engineer responsible for managing this data and making it usable, accessible, and trustworthy, rarely a day goes by without having to field some request from your stakeholders. But what happens when the data is wrong?

Have you ever been about to sign off after a long day running queries or building data pipelines only to get pinged by your head of marketing that "the data is missing" from a critical report? What about a frantic email from your CTO about "duplicate data" in a business intelligence dashboard? Or a memo from your CEO, the same one who is so bullish on data, about a confusing or inaccurate number in his latest board deck?

If any of these situations hit home for you, you're not alone.

This problem, often referred to as "data downtime," happens to even the most innovative and data-first companies, and, in our opinion, it's one of the biggest challenges facing businesses in the 21st century. Data downtime refers to periods of

time where data is missing, inaccurate, or otherwise erroneous, and it manifests in stale dashboards, inaccurate reports, and even poor decision making.

The root of data downtime? Unreliable data, and lots of it.

Data downtime can cost companies upwards of millions of dollars per year, not to mention customer trust. In fact, ZoomInfo found in 2019 that one in five companies lost a customer due to a data quality issue.

As you're likely aware, your company's bottom line isn't the only thing that's suffering from data downtime. Handling data quality issues consumes upwards of 40% of your data team's time that could otherwise be spent working on more interesting projects or actually innovating for the business.

This statistic probably comes as no surprise to you. It certainly didn't to us.

In a former life, Barr Moses served as VP of Operations at a customer success software company. Her team was responsible for managing reporting for the broader business, from generating dashboards for her CEO to use during All Hands meetings to setting strategy to reduce customer churn based on user metrics. She was responsible for managing her company's data operations and making sure stakeholders were set up for success when working with data.

Barr will never forget the day she came back to her desk from a grueling, hours-long planning session to find a sticky note with the words "The data is wrong" on her computer monitor. Not only was this revelation embarrassing; unfortunately, it also wasn't uncommon. Time and again she and her team would encounter these silent and small, but potentially detrimental, issues with their data.

There had to be a better way.

Poor data quality and unreliable data have been problems for organizations for decades, whether it's caused by poor reporting, false information, or technical errors. And as organizations increasingly leverage data and build more and more complex data ecosystems and infrastructure, this problem is only slated to increase.

The concept of "bad data" and poor data quality has been around nearly as long as humans have existed, albeit in different forms. With Captain Robert Falcon Scott and other early Antarctic explorers, poor data quality (or rather, data-uninformed decision making) led them to inaccurately forecast where and how long it would take to get to the South Pole, their target destination.

Several in more recent memory stick out, too. Take the infamous Mars Climate Orbiter crash in 1999. A NASA space probe, the Mars Climate Orbiter crashed as a result of a data entry error that produced outputs in non-SI (International System) units versus SI units, bringing it too close to the planet. This crash cost NASA a whopping $125 million. Like spacecraft, analytic pipelines can be extremely

vulnerable to the most innocent changes at any stage of the process. And this just scratches the surface.

Barr's unfortunate sticky note incident got her thinking: "I can't be alone!" Alongside Lior Gavish, Barr set out to get to the root cause of the "data downtime" issue. Together, they interviewed hundreds of data teams about their biggest problems, and time and again, data quality sprang to the top of the list. From ecommerce to healthcare, companies across industries were facing similar problems: schema changes were causing data pipelines to break, row or column duplicates were surfacing on business critical reports, and data would go missing in dashboards, causing them significant time, money, and resources to fix. We also realized that there needed to be a better way to communicate and address data quality issues as part of an iterative cycle of improving data reliability—and building a culture around driving data trust.

These conversations inspired us to write this book to convey some of the best practices we've learned and developed related to managing data quality at each stage of the data pipeline, from ingestion to analytics, and share how data teams in similar situations may be able to prevent their own data downtime.

For the purpose of this book, "data in production" refers to data from source systems (like CRMs, CMSs, and databases from any of the other analogies previously mentioned) that has been ingested by your warehouse, data lake, or other data storage and processing solutions and flows through your data pipeline (extract-transform-load, or ETL) so that it can be surfaced by the analytics layer to business users. Data pipelines can handle both batch and streaming data, and at a high level, the methods for measuring data quality for either type of asset are much the same.

Data downtime draws corollaries to software engineering and developer operations, a world in which application uptime or downtime (meaning, how frequently your software or service was "available" or "up" or "unavailable" or "down") is measured scrutinously to ensure that software is accessible and performant. Many site reliability engineers use "uptime" as a measurement because it correlates directly to the customer impact of poor software performance on the business. In a world where "five nines" (in other words, 99.999% uptime) of reliability is becoming the industry standard, how can we apply this to data?

In this book, we will address how modern data teams can build more resilient technologies, teams, and processes to ensure high data quality and reliability across their organizations.

In this chapter, we'll start by defining what data quality means in the context of this book. Next, we'll frame the current moment to better understand why data quality is more important for data leaders than ever before. And finally, we'll take a closer look at how best-in-class teams can achieve high data quality at each stage of the data pipeline and what it takes to maintain data trust at scale. This book focuses

primarily on data quality as a function of powering data analytical data pipelines for building decision-making dashboards, data products, machine learning (ML) models, and other data science outputs.

What Is Data Quality?

Data quality as a concept is not novel—"data quality" has been around as long as humans have been collecting data!

Over the past few decades, however, the definition of data quality has started to crystallize as a function of measuring the reliability, completeness, and accuracy of data as it relates to the state of what is being reported on. As they say, you can't manage what you don't measure, and high data quality is the first stage of any robust analytics program. Data quality is also an extremely powerful way to understand whether your data fits the needs of your business.

For the purpose of this book, we define data quality as the health of data at any stage in its life cycle. Data quality can be impacted at any stage of the data pipeline, before ingestion, in production, or even during analysis.

In our opinion, data quality frequently gets a bad rep. Data teams know they need to prioritize it, but it doesn't roll off the tongue the same way "machine learning," "data science," or even "analytics" does, and many teams don't have the bandwidth or resources to bring on someone full time to manage it. Instead, resource-strapped companies rely on the data analysts and engineers themselves to manage it, diverting them from projects that are perceived to be more interesting or innovative.

But if you can't trust the data and the data products it powers, then how can data users trust your team to deliver value? The phrase, "no data is better than bad data" is one that gets thrown around a lot by professionals in the space, and while it certainly holds merit, this often isn't a reality.

Data quality issues (or, data downtime) are practically unavoidable given the rate of growth and data consumption of most companies. But by understanding how we define data quality, it becomes much easier to measure and prevent it from causing issues downstream.

Framing the Current Moment

Technical teams have been tracking—and seeking to improve—data quality for as long as they've been tracking analytical data, but only in the 2020s has data quality become a top-line priority for many businesses. As data becomes not just an output but a financial commodity for many organizations, it's important that this information can be trusted.

As a result, companies are increasingly treating their data like code, applying frameworks and paradigms long-standard among software engineering teams to their data organizations and architectures. Development operations (DevOps), a technical field dedicated to shortening the systems development life cycle, spawned industry-leading best practices such as site reliability engineering (SRE), CI/CD (continuous integration / continuous deployment), and microservices-based architectures. In short, the goal of DevOps is to release more reliable and performant software through automation.

Over the past few years, more and more companies have been applying these concepts to data in the form of "DataOps." DataOps refers to the process of improving the reliability and performance of your data through automation, reducing data silos and fostering quicker, more fault-tolerant analytics.

Since 2019, companies such as Intuit, Airbnb, Uber, and Netflix have written prolifically about their commitment to ensuring reliable, highly available data for stakeholders across the business by applying DataOps best practices. In addition to powering analytics-based decision making (i.e., product strategy, financial models, growth marketing, etc.), data produced by these companies powers their applications and digital services. Inaccurate, missing, or erroneous data can cost them time, money, and the trust of their customers.

As these tech behemoths increasingly shed light on the importance and challenges of achieving high data quality, other companies of all sizes and industries are starting to take note and replicate these efforts, from implementing more robust testing to investing in DataOps best practices like monitoring and data observability.

But what has led to this need for higher data quality? What about the data landscape has changed to facilitate the rise of DataOps, and as such the rise of data quality? We'll dig into these questions next.

Understanding the "Rise of Data Downtime"

With a greater focus on monetizing data coupled with the ever-present desire to increase data accuracy, we need to better understand some of the factors that can lead to data downtime. We'll take a closer look at variables that can impact your data next.

Migration to the cloud

Twenty years ago, your data warehouse (a place to transform and store structured data) probably would have lived in an office basement, not on AWS or Azure. Now, with the rise of data-driven analytics, cross-functional data teams, and most importantly, the cloud, cloud data warehousing solutions such as Amazon Redshift, Snowflake, and Google BigQuery have become increasingly popular options for

companies bullish on data. In many ways, the cloud makes data easier to manage, more accessible to a wider variety of users, and far faster to process.

Not long after data warehouses moved to the cloud, so too did data lakes (a place to transform and store unstructured data), giving data teams even greater flexibility when it comes to managing their data assets. As companies and their data moved to the cloud, analytics-based decision making (and the need for high-quality data) became a greater priority for businesses.

More data sources

Nowadays, companies use anywhere from dozens to hundreds of internal and external data sources to produce analytics and ML models. Any one of these sources can change in unexpected ways and without notice, compromising the data the company uses to make decisions.

For example, an engineering team might make a change to the company's website, thereby modifying the output of a data set that is key to marketing analytics. As a result, key marketing metrics may be wrong, leading the company to make poor decisions about ad campaigns, sales targets, and other important, revenue-driving projects.

Increasingly complex data pipelines

Data pipelines have become increasingly complex with multiple stages of processing and nontrivial dependencies between various data assets as a result of more advanced (and disparate) tooling, more data sources, and increasing diligence afforded to data by executive leadership. Without visibility into these dependencies, however, any change made to one data set can have unintended consequences impacting the correctness of dependent data assets.

In short, a lot goes on in a data pipeline. Source data is extracted, ingested, transformed, loaded, stored, processed, and delivered, among other possible steps, with many APIs and integrations between different stages of the pipeline. At each juncture, there's an opportunity for data downtime, just like there's an opportunity for application downtime whenever code is merged. Additionally, things can go wrong even when data isn't at a critical juncture, for instance, when data is migrated between warehouses or manually entered in a source system.

More specialized data teams

As companies increasingly rely on data to drive smart decision making, they are hiring more and more data analysts, data scientists, and data engineers to build and maintain the data pipelines, analytics, and ML models that power their services and products, as well as their business operations.

While data analysts are primarily responsible for gathering, cleaning, and querying data sets to help functional stakeholders produce rich, actionable insights about the business, data engineers are responsible for ensuring that the underlying technologies and systems powering these analytics are performant, fast, and reliable. In industry, data scientists typically collect, wrangle, augment, and make sense of unstructured data to improve the business. The distinction between data analysts and data scientists can be a little vague, and titles and responsibilities often vary depending on the needs of the company. For instance, in the late 2010s, Uber changed all data analysts' titles to data scientists after an organizational restructure.

As data becomes more and more foundational to business, data teams will only grow. In fact, larger companies may support additional roles including data stewards, data governance leaders, operations analysts, and even analytics engineers (a hybrid data engineer-analyst role popular with startups and mid-sized companies who may not have the resources to support a large data team).

With all of these different users touching the data, miscommunication or insufficient coordination is inevitable and will cause these complex systems to break as changes are made. For example, a new field added to a data table by one team may cause another team's pipeline to fail, resulting in missing or partial data. Downstream, this bad data can lead to millions of dollars in lost revenue, erosion of customer trust, and even compliance risk.

Decentralized data teams

As data becomes central to business operations, more and more functional teams across the company have gotten involved in data management and analytics to streamline and speed up the insights gathering process. Consequently, more and more data teams are adopting a distributed, decentralized model that mimics the industry-wide migration from monolithic to microservice architectures that took the software engineering world by storm in the mid-2010s.

What is a decentralized data architecture? Not to be confused with the data mesh, which is an organizational paradigm that leverages a distributed, domain-oriented design, a decentralized data architecture is managed by a central data platform team, with analytical and data science teams distributed across the business. Increasingly, we're finding that more and more teams leaning into the embedded data analyst model are relying on this type of architecture.

For instance, your 200-person company may support a team of 3 data engineers and 10 data analysts, with analysts distributed across functional teams to better support the needs of the business. These analysts will report to operational teams or centralized data teams but own specific data sets and reporting functions. Multiple domains will generate and leverage data, leading to the inevitability that data sets used by multiple teams become duplicated, go missing, or go stale over time. If you're

reading this book, you're probably no stranger to the experience of using a data set that's no longer relevant, unbeknownst to you!

Other Industry Trends Contributing to the Current Moment

In addition to the aforementioned factors that frequently lead to data downtime, several industry shifts are also occurring as a result of technological innovation that are driving transformation of the data landscape. These shifts are all contributors to this heightened attention to data quality.

Data mesh

Much in the same way that software engineering teams transitioned from monolithic applications to microservice architectures, the data mesh is, in many ways, the data platform version of microservices. It's important to note that the concept of data mesh is nascent, and there is much discussion in the data community regarding how (or whether it makes sense) to execute on one at both a cultural and technical level.

As first defined by Zhamak Dehghani, a Thoughtworks consultant and the original architect of the term, a data mesh, illustrated in Figure 1-1, is a sociotechnical paradigm that recognizes the interactions between people and the technical architecture and solutions in complex organizations. The data mesh embraces the ubiquity of data in the enterprise by leveraging a domain-oriented, self-serve design. It utilizes Eric Evans's theory of domain-driven design, a flexible, scalable software development paradigm that matches the structure and language of your code with its corresponding business domain.

Unlike traditional monolithic data infrastructures that handle the consumption, storage, transformation, and output of data in one central data lake, a data mesh supports distributed, domain-specific data consumers and views "data-as-a-product," with each domain handling their own data pipelines. The tissue connecting these domains and their associated data assets is a universal interoperability layer that applies the same syntax and data standards.

Data meshes federate data ownership among domain data owners who are held accountable for providing their data as products, while also facilitating communication between distributed data across different locations.

While the data infrastructure is responsible for providing each domain with the solutions with which to process it, domains are tasked with managing ingestion, cleaning, and aggregation of the data to generate assets that can be used by business intelligence applications. Each domain is responsible for owning their pipelines, but there is a set of capabilities applied to all domains that stores, catalogs, and maintains access controls for the raw data. Once data has been served to and transformed by