LEARNING MADE EASY

Snowflake Special Edition

Cloud Data Engineering

for dummies®
A Wiley Brand

What is modern data engineering?

How it powers operational and advanced data analytics

How to achieve data engineering success

Brought to you by

snowflake™

David Baum

About Snowflake

Snowflake's cloud data platform shatters barriers that have prevented organizations of all sizes from unleashing the true value from their data. Thousands of customers deploy Snowflake to advance their businesses beyond what was once possible by deriving insights from their data by all their business users. Snowflake equips organizations with a single, integrated platform that offers the data warehouse built for the cloud; instant, secure, and governed access to their network of data; and a core architecture to enable many types of data workloads, including a single platform for developing modern data applications. Snowflake: Data without limits.

Find out more at **Snowflake.com.**

Cloud Data Engineering

Snowflake Special Edition

by David Baum

for dummies®
A Wiley Brand

Cloud Data Engineering For Dummies®, Snowflake Special Edition

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

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, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. For information about licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

ISBN 978-1-119-75456-5 (pbk); ISBN 978-1-119-75458-9 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

We're proud of this book and of the people who worked on it. Some of the people who helped bring this book to market include the following:

Development Editor: Colleen Diamond

Project Editor: Martin V. Minner

Editorial Manager: Rev Mengle

Executive Editor: Steve Hayes

Business Development Representative: Karen Hattan

Production Editor: Siddique Shaik

Snowflake Contributors Team: Vincent Morello, Shiyi Gu, Kent Graziano, Clarke Patterson, Jeremiah Hansen, Mike Klaczynski, Dinesh Kulkarni, Leslie Steere

Table of Contents

Introduction

If data is the heartbeat of the enterprise, then data engineering is the activity that ensures current, accurate, and high-quality data is flowing to the solutions that depend on it. As analytics become progressively more important, data engineering has become a competitive edge and central to the technology initiatives that position companies for success.

Data engineers oversee the ingestion, transformation, delivery, and movement of data throughout every part of an organization. They extract it from many sources. They load it into data warehouses and data lakes. They transform it for data science initiatives, massage it for predictive analytic systems, and deliver it for business intelligence dashboards.

Today, many data engineering activities are moving from highly technical data professionals into the hands of business users. As IT cedes control of some types of data, user-friendly data engineering tools give business users more independence.

For example, data scientists need massive amounts of data to develop and train machine learning models, which constantly churn through new data to automate processes and make predictions. Marketing professionals amass data from weblogs, clickstreams, mobile applications, and social media networks to understand customer behavior. Retailers combine point-of-sale data with weather data to forecast consumer demand and automate supply chains.

In this book, you'll discover how your business and IT teams can effectively collaborate as the entire organization adopts modern data engineering tools and procedures. You'll see how IT teams lay the data engineering foundation to build reliable, high-performance data pipelines that can benefit the entire organization. You'll also learn how self-service data preparation and integration activities can be extended to analysts, data scientists, and line-of-business users. Lastly, you'll discover how a cloud data platform enables your organization to amass all of its data in one central location, making it easy to consolidate, cleanse, transform, and deliver that data to a wide range of analytic tools and securely share it with internal and external data consumers.

Who Should Read This Book

Cloud Data Engineering For Dummies, Snowflake Special Edition, explains why data engineering is important and introduces the essential components of a modern data engineering practice. Professionals of all levels of technical proficiency can benefit from this book:

- **ETL developers** who are familiar with traditional extract, transform, and load (ETL) technologies can learn about the modern practice of ingesting, transforming, and delivering data.
- **Data engineers** who already work with the latest technologies can discover a broader context for their activities.
- **IT and data leaders** who want to establish more effective data engineering practices can see how to automate labor-intensive data preparation procedures.

Icons Used in This Book

This book uses the following icons to help you more easily find the information you need:

TIP

Paragraphs marked with this icon show faster ways to perform essential tasks.

REMEMBER

This icon highlights ideas worth remembering as you immerse yourself in data engineering concepts.

CASE STUDY

Look for this icon to read about organizations that have successfully applied modern data engineering practices and principles.

Beyond the Book

Visit `www.snowflake.com` to find additional content about data engineering and related topics, get in touch with Snowflake, or try Snowflake for free.

IN THIS CHAPTER

- **Understanding the basics of data engineering**
- **Knowing the history of data engineering**
- **Recognizing the benefits of serverless computing**
- **Shifting data responsibilities from IT to business users**

Chapter 1
Charting the Rise of Modern Data Engineering

The software industry is driven by innovation, but most new technologies have historical precedents. This chapter describes the data engineering principles, practices, and capabilities that have paved the way for breakthroughs in data management and analytics.

Understanding How Data Engineering Works

Data engineering encompasses a broad set of procedures, tools, and skill sets that govern and facilitate the flow of data. *Data engineers* are experts at making data ready for consumption by working with multiple systems and tools. They may include a series of enterprise and cloud applications; a data warehouse; a data lake environment; continuous streams of data from Internet of Things (IoT) sensors; or many other types of databases, applications, and information systems. Data engineers make an organization's data "production ready" by putting it into a usable form, and typically in a centralized repository or cloud data platform. They understand how to manipulate data formats, scale data systems, and enforce data quality and security.

Data engineers are focused primarily on building and maintaining *data pipelines* that transport data through different steps and put it into a usable state. Their charter is to ensure the data is clean, reliable, and properly transformed for whatever the user community requires, including running queries for front-line workers, generating reports for management, or training machine learning models for data scientists. They are skilled at wrangling data from hundreds of sources, including enterprise applications, software-as-a-service (SaaS) solutions, and weblog data, that generate immense amounts of data in unique formats.

The data engineering process encompasses the overall effort required to create data pipelines that automate the transfer of data from place to place and transform that data into a specific format for a certain type of analysis (see Figure 1-1). In that sense, data engineering isn't something you do once. It's an ongoing practice that involves collecting, preparing, transforming, and delivering data. A data pipeline helps automate these tasks so they can be reliably repeated. It's a practice more than a specific technology.

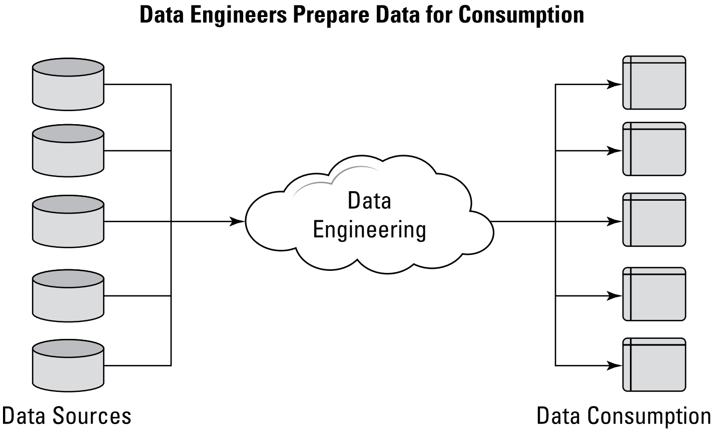

FIGURE 1-1: Data engineering encompasses a broad range of activities related to collecting data and then putting it into a useful form for consumption by analysts, data scientists, customers, and partners.

Reviewing the History of Data Engineering

Extract, transform, and load (ETL), or what we now call data engineering, used to be much simpler. There was much less data in the world, of fewer types, and needed at a much slower pace. Enterprise data was moved from one system to another, and software professionals generally transmitted it in *batch mode,* as a bulk data load operation. When data needed to be shared, it was often moved through File Transfer Protocol (FTP), application programming interfaces (APIs), and web services.

ETL procedures orchestrated the movement of data and converted it into a common format. These tasks were mostly handled by the IT department, in response to specific requests. Business users would identify their data requirements, and the IT team would see to it the data was placed into a format and location to fulfill those requirements.

Often, the source of this data was a small number of enterprise business applications such as enterprise resource planning (ERP), supply chain management (SCM), and customer relationship management (CRM) systems. The destination was a data warehouse, where it was loaded into highly structured tables from which it could be readily accessed via Structured Query Language (SQL) tools. This data was predictable, manageable, and generally arising from a handful of sources. Because the data was already structured, transformation needs were straightforward and much simpler. The ETL process was repeated once per day, week, or month, depending on the latency requirements of the business application or analytics use case. This typically involved some fairly straightforward transformations, perhaps aided by a data integration tool that helped automate the ETL process.

Explaining the Impact of New Technology Paradigms

New technologies constantly pave the way for new types of data management and analysis. Innovation is still hard at work doubling the power and capacity of computers every 18 months or so. The cost of storing data has gone down significantly, even as the computing devices that process that data have become more powerful. As the cloud computing industry matures, a growing number of organizations continue to use cloud services to store, manage, and process their data. The cloud offers virtually unlimited capacity and near-infinite elasticity and scalability, allowing companies of any size to deploy a large number of concurrent, high-performance workloads within a centralized platform.

Having affordable, scalable, and elastic storage options enables new types of analysis — and necessitates new methods of data engineering. With lower costs and faster computing infrastructure, you can now retain years of historical data and gradually uncover patterns and insights from that data. You don't need to know all the questions in advance, or even how to transform the data before it's loaded.

Previously, businesses had to be selective about which data they collected and stored. Now, advanced data repositories such as data lakes, data warehouses, and cloud data platforms allow businesses to capture the data first and ask questions later. For example, data scientists often want data in its raw form to ensure they don't lose any of the information. They want to maximize insights that can be derived from that data. They don't necessarily need to figure out what they're going to do with the data ahead of time or even how to store, structure, or format it. This gives data scientists the luxury of casting a wide, exploratory net. Once they find patterns in the data, they can determine how to best use it.

This shift from seeking answers to specific questions to first exploring data in a much broader sense, a process known as *data discovery*, uncovers new realms of inquiry. Initially, these "big data" exploration projects were conducted using Apache Hadoop, an open source software framework that distributes data storage and processing among commodity hardware located in on-premises data centers. Unfortunately, many of these on-premises data lake projects failed due to their complexity, slow time to value, and heavy system management efforts.

Newer cloud paradigms have largely replaced these legacy Hadoop environments, partly because of the cloud's more cost-effective and efficient compute and storage options.

The cloud has also given rise to highly efficient methods of application development and operations (DevOps). These include *serverless computing*, in which the cloud service provider automatically provisions, scales, and manages the infrastructure required to host your data and run your business applications. A serverless environment allows developers to bring products to market faster via *containers* (software packages that contain everything needed to run an application), *microservices* (applications built as modular components or services), and *continuous integration/continuous delivery* (CI/CD) processes. Chapter 2 discusses these concepts further.

Moving Business Users into the Driver's Seat

Data is everywhere. Discovering new ways to combine, analyze, and share that data continues to drive operational efficiencies and uncover new business opportunities. It also changes the basic responsibilities surrounding data engineering.

In the past, when constructing traditional ETL pipelines, business users may have met once or twice with the IT team to discuss the requirements. Now, business users often have a better understanding of what that data means and what value it holds. They have become the *content stakeholders*, and many of them are eager to do their own data preparation and exploration.

What is the profile of my data? How can I tell if the quality is good enough for the types of analysis I want to perform? These are common questions for which business users seek immediate answers.

In response, some data integration activities are moving away from IT and into the arms of the business community. These are largely positive trends, but caution is in order: Organizations must keep track of who works with the data and what changes are made.

Organizations must also discourage the practice of producing multiple copies of the data, because having more than one version can lead to discrepancies later on, and a data management nightmare. Even the smallest data discrepancy can produce a reporting delta between two business units that equates to millions or billions of dollars in revenue, costs, and profits. As Chapter 4 shows, data governance must be the foundation of these "citizen integration" efforts.

TIP

Democratizing data engineering activities extends more responsibility to business users. Although you want your data engineering practice to be agile and accessible to people throughout the organization, you also have to pay attention to good data governance. IT must work closely with the business to ensure all data security, compliance, and regulatory policies are met.

DATA INGESTION WITH A CLOUD DATA PLATFORM

CASE STUDY

Fair is a financial technology company that offers a new way to shop for a car, get approved for a loan, and pay for the car. Its unique smartphone app gives customers the freedom to drive the car they want for as long as they want and gives them the flexibility to turn in the vehicle at any time. Data is essential to this business model: The company ingests and analyzes billions of data points from more than 500 data sources.

Previously, Fair's legacy data warehouse could not keep pace with the company's rapidly expanding appetite for data. This led to frequent contention for scarce server resources, resulting in the creation of *data marts* — siloed copies of some of the data stored in a data warehouse to offload analytic activity and preserve performance. In addition, dealer inventory data was imported only once per day to avoid system overload, making real-time analytics impossible. And Fair's analytics team spent hours troubleshooting cluster failures and waiting for ETL jobs to run.

Seeking to enable real-time analytics for data science, marketing, and operational reporting, Fair subscribed to a cloud data platform-as-a-service that could easily ingest, analyze, and query all its data, including semi-structured JavaScript Object Notation (JSON) data from tables that contain billions of rows. The cloud data platform enables new data pipelines that ingest dealer inventory data continuously. ETL jobs, which previously required several hours to run, now execute in five minutes or less. The platform can ingest the JSON data in its native format. And having a cloud architecture that separates storage and compute resources eliminates resource contention for data engineering workloads, enabling marketers and data scientists to extract minute-by-minute insights from the data.

IN THIS CHAPTER

- Understanding how a data pipeline works
- Collecting and ingesting data
- Transforming data into business-ready formats
- Sharing and consuming shared data across your business and beyond

Chapter 2 Describing the Data Engineering Process

Data engineering involves ingesting, transforming, delivering, and sharing data for analysis. These fundamental tasks are completed via data pipelines that automate the process in a repeatable way. This chapter describes the primary procedures that make this possible.

Understanding How Modern Data Pipelines Work

A *data pipeline* is a set of data-processing elements that move data from source to destination, and often from one format (raw) to another (analytics-ready).

Most modern pipelines use three basic steps. The first step is collection, during which raw data is loaded into a repository or data platform, often in a *raw data zone*. In the second step, transformation, the data is standardized, cleansed, mapped, or combined with data from other sources. Transformation also entails modifying the data from one data type to another, so it is ready for different types of data consumption. And finally, data delivery and secure data sharing makes business-ready data available to other users and departments, both within the organization and externally (see Figure 2-1).

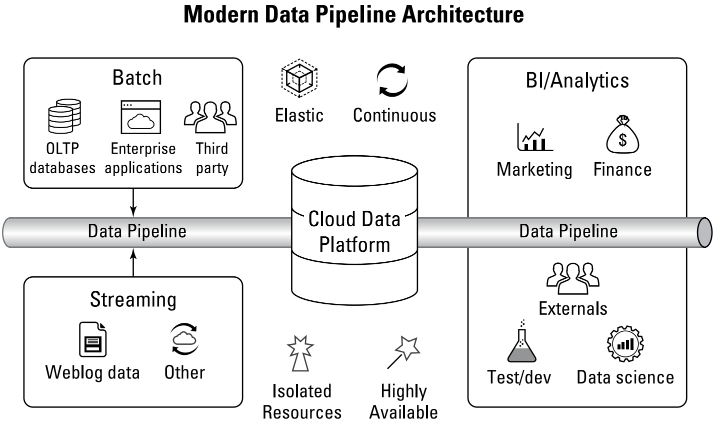

FIGURE 2-1: A modern data pipeline accommodates many types of data, transmission methods, and analytic use cases.

WHAT TO LOOK FOR IN A DATA PIPELINE

Your data pipeline should have the following characteristics:

Scalable performance. The ability to ingest and transform data without impacting performance or experiencing resource contention. For example, you should be able to ramp up data integration workloads without affecting the performance of analytical workloads in the data warehouse or data lake.

Extensible but based on standards. Data engineers should be able to choose from a variety of languages and tools. For example, some may use Structured Query Language (SQL) but also want to enable extensibility with Java, Python, and other languages and tools.

Batch and streaming data. A complete data pipeline should support a range of data ingestion and integration options, including batch data loading and replication, as well as streaming ingestion and integration.

Collecting and Ingesting Data

Many types of data exist, and you can store it in many ways, on premises and in the cloud. For example, your business may generate data from transactional applications, such as customer relationship management (CRM) data from Salesforce or enterprise resource planning (ERP) data from SAP. Or you may have Internet of Things (IoT) sensors that gather readings from a production line or factory floor operation.

Legacy data integration solutions do a good job of ingesting highly structured and batch data, but they are often too rigid to collect and ingest newer types of data, such as machine-generated data from IoT systems, streaming data from social media feeds, and weblog data from Internet and mobile apps. The modern approach allows you to easily and efficiently ingest all these types of data, and it should support a range of popular data ingestion styles including batch integration and streaming integration via streaming processing systems such as Apache Kafka.

Data collection is a time-consuming task. A typical marketing department, for example, may depend on 20 different SaaS applications for online advertising, web analytics, marketing operations, and so on. Coding discrete interfaces to collect data from all these apps can be a huge job, which is why most organizations use data ingestion tools to automate the process. These tools shield developers from having to create and document custom APIs.

This type of flexibility is paramount to gaining immediate value from your data, especially if streaming data needs to be addressed. With streaming data, there is no start or end to the data: It simply comes in as a stream. You don't always know when it will be coming, but you know it needs to be captured.

REMEMBER

Whether data arises from an enterprise application, a website, or a series of IoT sensors, data engineers must figure out how to connect to those data sources, collect the data in reliable ways, and ingest it into a single repository, with the final goal of making it accessible and useful to the user community. On the flip side, traditional legacy architectures often require you to land and transfer data into many systems, which can lead to having many different copies of your data in many different places. It's much simpler to manage your data when you can consolidate it all in one location as a single source of truth.

Transforming Data

Data transformation is the process of preparing data for different kinds of consumption. It can involve *standardizing* (converting all data types to the same format), *cleansing* (resolving inconsistencies and inaccuracies), *mapping* (combining data elements from two or more data models), *augmenting* (pulling in data from other sources), and so on.

You can choose from a range of data transformation options to suit the skills and preferences of your user base. Tech-savvy developers may build their own transformation logic, using their favorite programming languages and integrated development environments (IDEs). This often entails having knowledge of many open source systems and technologies. Others may prefer to use a graphical drag-and-drop interface such as those from Talend, Informatica, or Matillion to design data pipelines. These tools abstract the complexity of the underlying data protocols and help organizations build transformation logic faster.

Designing pipelines

How data pipelines are designed has everything to do with the underlying database and processing engine as well as the skill sets of your team. These design decisions invariably reflect the choice of underlying architecture.

For example, suppose you used Hadoop as the file system for your data lake, and your data pipelines were based on MapReduce. Only a few years later, you decide to leverage Spark as a much faster processing framework, requiring you to modify or re-create all your pipelines accordingly.

Accommodating these technology transitions is much easier if you have a reliable data platform that simplifies your architecture so you don't need to stitch everything together by yourself. This gives you more options as you experiment with new use cases and your technology environment evolves.

Data transformation has many critical aspects, but chief among them is data quality. Data must be cleansed to eliminate errors and duplications, such as variant name spellings, addresses, and other anomalies. Good data quality ensures accurate reporting and analytics, and ultimately allows for advanced use cases such as machine learning.

Evolving from ETL to ELT

As stated in Chapter 1, *ETL* refers to the process of extracting, transforming, and loading data. With ETL, data needs to be transformed outside of the target system and uses a separate processing engine, which involves unnecessary data movement and changes, and tends to be slow. Modern data integration workloads are enhanced by leveraging the processing power of target databases. In these instances, the data pipelines are designed to *extract* and *load* the data first, and then *transform* it later (ELT).

ELT is especially conducive to advanced analytics. For example, data scientists commonly load data into a data lake and then combine it with another data source, or use it to train predictive models. Maintaining the data in a raw (or less processed) state allows data scientists to keep their options open. This approach is quicker, because it leverages the power of modern data processing engines and cuts down on unnecessary data movement.

REMEMBER

Efficient ELT does not require things such as data schema at the outset, even for semi-structured data. Data can simply be loaded in the raw form and transformed later, once it is clear how it will be used.

Delivering and Sharing Data

Data delivery is the act of providing data to authorized users, workgroups, and departments within an organization or externally to customers and business partners. Traditional data delivery methods use APIs, File Transfer Protocol (FTP), or file services to copy data and send it to consumers. However, none of these methods is optimal for securely sharing data, mainly because the data quickly becomes stale and must be constantly refreshed with more current versions, requiring regular data movement. Too many copies quickly become a governance and security nightmare.

Modern data sharing technologies enable organizations to easily share access to read-only slices of their data, and to receive shared data, in the same secure and governed way. They can help you avoid data movement, eliminate ETL processes, and minimize the use of constant update procedures to keep data current. With these technologies, you don't need to transfer data via FTP or configure APIs to link applications.

Modern data sharing involves simply granting access to live, governed, read-only data by pointing at its original location. With granular-level access control, data is shared rather than copied, and no additional cloud storage is required. With this more advanced architecture, data providers can easily and securely publish data for instant discovery, query, and enrichment by data consumers, as shown in Figure 2-2.

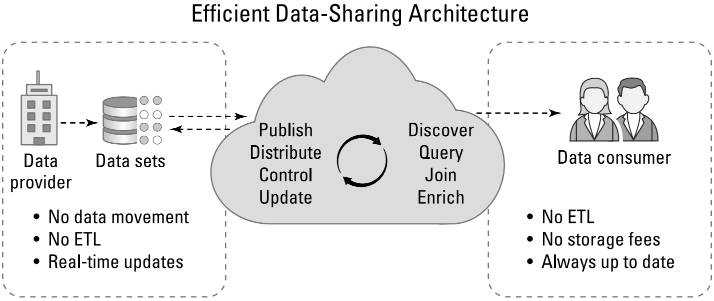

FIGURE 2-2: A data pipeline architecture for near real-time data sharing.

Some modern cloud data platforms are designed to facilitate this type of secure data sharing. Your organization maintains one copy of data and provides governed *access* to it by authorized users, both internal and external, and delivers updates to that data in near real time.

Whatever industry or market you operate in, having all your data on hand and readily accessible opens doors to new opportunities. This is especially true when that data is stored and managed in a consistent way, and when you can use the near-infinite capacity of the cloud to scale your data initiatives.

A cloud data platform allows you to store all your data in one place, in its raw form, regardless of format, and deliver analytics-ready data to the people who need it. It provides convenient access to that data and improves the speed at which you can ingest, transform, and share data across your organization — and beyond.

IN THIS CHAPTER

- Reaping the full benefits from cloud-based solutions
- Knowing why data latency matters
- Using tools to simplify data integration tasks
- Creating downstream analytics

Chapter 3 Mapping the Data Engineering Landscape

As you create data pipelines, remember the ultimate goal: to turn your data into useful information such as actionable analytics for business users and predictive models for data scientists. To do so, you must think about the journey your data will take through your data pipelines. Start by answering some fundamental questions:

- What business questions do you want to answer?
- What types of data will you be analyzing?
- What kinds of schema do you need to define?
- What types of data quality problems do you have?
- What is the acceptable latency of your data?
- Will you transform your data as you ingest it, or maintain it in a raw state and transform it later, for specific use cases?

Once you have answered these questions, you can determine what type of data pipeline you need, how frequently you need to update your data, and whether you should use data lakes, data warehouses, integration tools, or simply a cloud data platform to simplify the process of creating database interfaces, data ingestion procedures, and data transformation logic.

Working with Data Warehouses and Data Lakes

Data engineering involves extracting data from various applications, devices, event streams, and databases. Where will it land? For many companies, the answer is often a data warehouse or a data lake:

- **Data lakes** are scalable repositories that can store many types of data in raw and native forms, especially for semi-structured and unstructured data. To be truly useful, they must facilitate user-friendly exploration via popular methods such as SQL, automate routine data management activities, and support a range of analytics use cases.
- **Data warehouses** typically ingest and store only structured data, usually defined by a relational database schema. Raw data often needs to be transformed to conform to the schema. Modern data warehouses are optimized for processing thousands or even millions of queries per day, and can support specific business uses.

Today, cloud-built versions of these data storage solutions have given rise to hybrid options that combine the best attributes of both the data lake and data warehouse. Some of these newer options are designed to store and work with structured and semi-structured data as well as the most important attributes of unstructured data.

Capitalizing on Cloud Solutions

Although many storage and compute solutions have been moved to the cloud, not every cloud solution is equal. A true cloud-built and cloud-optimized architecture allows you to capitalize on the benefits of the cloud. These benefits include:

- Centralized storage for all data.
- Independent scaling of storage and dedicated computing resources.
- The ability to load and query data simultaneously without degrading performance.

True cloud data platforms are built using a cloud-optimized architecture that takes advantage of storage as a service, where data storage expands and contracts automatically.

In addition, your computing resources are virtually unlimited, so you can scale up and down, automatically or on the fly, to accommodate peak workloads, and pay only for the computing power you need on a per-second basis.

A modern cloud data platform can seamlessly replicate data across multiple regions and clouds to enhance business continuity. It can also simplify the storing, loading, integrating, scaling, and analysis of data for data-intensive workloads such as business intelligence (BI) reporting and machine learning.

Understanding Data Latency

Data latency is the time delay between when data is generated and when it is available for use. As mentioned in Chapter 1, in previous times, data was periodically loaded into analytic repositories in batches, generally daily, weekly, or monthly. Today, more analytic workloads require data that is updated in near real time, such as every five minutes, as well as streaming data that may be loaded continuously.

Consider a financial services company that has a data warehouse designed to store and analyze core banking data. Initially, only one daily ingest of this data was necessary, for historical reporting. But as the company began to place more emphasis on private banking services and brokerage accounts, financial advisors needed to see reports that reflected recent transactions. So the firm created a data pipeline that loads new transactions every few minutes, in conjunction with a predictive model that enables advisors to suggest relevant products and services based on current activity.

Although people commonly use the phrase *real-time data*, the definition and need for real-time data varies by use case: Live trading data may require streaming latency in milliseconds, while a store inventory report may tolerate a few minutes. Far more common are pipelines that need to handle *near real-time* data, in which data is refreshed every few minutes.

Change data capture (CDC) capabilities simplify data pipelines by recognizing the changes that have occurred since the last data load and incrementally processing or ingesting that data. For example, in the case of the financial services company, a bulk upload from the banking system refreshes the data warehouse each night, while CDC adds new transactions every five minutes. This type of process allows analytic databases to stay current without reloading the entire data set.

TIP

In the case of streaming data, be aware that *event time* and *processing time* are not always the same. You can't simply follow the timestamp in the data, since some transactions may be delayed in transit, which could cause them to be recorded in the wrong order. If you need to work with streaming data, you may need to create a pipeline that can verify precisely when each packet, record, or transaction occurred, and ensure they are recorded only once, and in the right order, according to your business requirements. Adding event time to the record ensures that processing delays do not cause incorrect results due to an earlier change overwriting a later change.

Automating Integration Tasks

Some software developers build APIs and integrated development environments (IDEs) with different programming languages (see Chapter 2). Less technical users and many data engineers may prefer to use a combination of off-the-shelf *integration tools* and *orchestration tools* to create integration logic, cleanse data, and schedule and orchestrate pipelines:

- **Data integration tools,** such as Informatica and Talend, provide platforms for managing the movement of data through your pipeline with data integration, data quality, application integration, and so on.
- **Data orchestration tools,** such as Apache Airflow, focus on orchestration only and structure the flow of data as a series of events. This open source workflow management tool keeps data flowing through the pipeline as defined by sequential tasks. Some commercial integration vendors also assist with scheduling and orchestration tasks.

Saving time for developers

Data integration platforms and tools sort out the diversity of data types and APIs so data engineers can connect to data sources directly instead of coding APIs, which is complex and time-consuming. For example, you may need to connect to a complex legacy SAP enterprise resource planning (ERP) system that requires remote function calls (RFCs), business application programming interfaces (BAPIs), and Intermediate Document (IDoc) messages. Using an integration tool to connect to a data source is much more efficient than building and managing your own.

Integration tools also include universal connectors and adapters to accommodate many types of data, and they handle all the necessary transformations automatically. Most of these tools also offer data integration workbenches where you can select data sources, destinations, and other variables via a point-and-click, drag-and-drop GUI environment.

Users also can visually create data pipelines and some customizable options. Data integration tools help them craft the necessary software interfaces. Instead of having to master the nuances of transformation logic, networking protocols, and data models, the integration tools encapsulate the nitty-gritty details of merging, mapping, and integrating your data.

Data orchestration tools help structure pipeline tasks such as scheduling jobs, executing workflows, and coordinating dependencies among tasks. Similar to data integration tools, they make it easier to access and combine data across storage systems, and they present that data via standardized APIs. Think of it as the "plug in the wall" concept: You don't worry about where your electricity comes from. You just plug in your appliances and they work. In this fashion, data integration tools and data orchestration tools "abstract out" the connection details and data access details to make many types of data accessible in a consistent way. They also help to "future proof" your analytic applications by allowing you to easily pull in new types of data.

Deciding whether to "build or buy"

To determine whether to build integrations from scratch or purchase one or more tools, start by examining the skills and capacity of your team. How many types of data will you be working with, and how many software interfaces do you need to establish? How many engineering resources are you willing to spend on not only building but also maintaining data pipelines? You may have both simple transformations and highly complex and customized logic. Does it make more sense to use your skilled data engineers to hand-code everything, or to purchase a general-purpose integration tool your data engineers can use and customize?

Organizations can choose from a variety of processing engines and programming languages to build data pipelines. For example, Apache Spark, Flink, and Beam are open source projects for processing frameworks. You can also use a cloud data platform with built-in processing power to simplify architecture and maintenance. In addition, consider how extensible your pipelines can be so your team can use the language of their choice, whether it’s SQL, Java, Python, or Scala.

Confronting Data Preparation Challenges

Many BI tools today enable ad hoc analysis by business users. Drag-and-drop environments make it easy for them to visualize data, populate dashboards, and generate reports without coding. But even as these modern analytic tools have created new value, they have also exposed bottlenecks associated with accessing and preparing data for analysis. Analytical initiatives will succeed only if the right data is delivered at the right time, in the correct form, which lies mainly with the data preparation processes. This is especially true for advanced analytics such as machine learning.

Machine learning is an iterative and data-intensive activity that moves data through multiple stages, from discovery and development into production. The success of each model depends on the data preparation process in which large volumes of diverse data must be collected, cleansed, transformed, and reformatted in many ways.

Data scientists experiment with many data sets as they create and train machine learning models. A model designed to predict customer churn, for example, may incorporate data about customer behavior relative to sales, service, and purchasing, both historic and current. Each time data scientists pull in new data, they must wait for data engineers to load and prepare the data set, which introduces latency into the flow.

They also must rescale the data into a range or with specific requirements (often referred to as *normalization* and *standardization*), as required by each machine learning algorithm. Machine learning models must be periodically retrained, which requires fresh data to be reprocessed through the cycle, often via manual extract, transform, and load (ETL) processes that can potentially introduce errors.

All these data preparation challenges are easier to resolve with a cloud data platform that centralizes data and supports modern data pipeline development.

Centralizing Data in the Cloud

Rather than requiring a data warehouse for BI and a data lake for data science initiatives, a cloud data platform serves as a unified, highly accessible repository that serves both purposes, including the entire data lifecycle of machine learning, artificial intelligence, and predictive application development.

Such a platform can accommodate raw, structured, semi-structured, and the crucial elements of unstructured data to maximize options for business users, analysts, and data scientists. Instead of drawing data from multiple data warehouses or data lakes, the platform stores data in one scalable repository, providing a single source of truth based on cleansed and synchronized data. It also brings together raw and transformed data and makes both available for data pipelines that serve various analytics needs.

A flexible cloud data platform accommodates batch and streaming data. You can land data in one place and leverage various languages to transform, prepare, and query it to support different requirements. Many cloud data platforms support open source languages such as SQL, Spark, Python, and Java to transform and prepare data. One benefit of being able to access semi-structured data via SQL is the ability to create ELT pipelines without having to first define the data schema. This is especially advantageous for JSON and XML data, which may change often in cloud application sources.

To simplify data pipeline development, look for a data platform that also has a built-in data ingestion service designed to asynchronously load data into the cloud storage environment. You'll also want support for CDC technology, so when data is changed, detecting the differences is easy. This increases performance by allowing you to work with the changed data instead of bringing in the whole data set again.

RESPONSIVE DATA PIPELINE

CASE STUDY

As one of Australasia's leading marketplace lending platforms, Harmoney has matched more than 32,000 borrowers with more than $1 billion in personal loans. Its lending experience is 100 percent online, from application to funding, necessitating complete and accurate data collection, processing, and analysis.

Harmoney's previous data warehouse, implemented in a traditional database solution, could not ingest certain file types or natively connect to a range of cloud-based analytics systems and third-party data services. Its data pipeline used custom-coded Python connectors, which were rigid and hard to maintain.

To speed data-ingestion activities and accommodate new types of data, Harmoney adopted a cloud data platform designed to store vast amounts of data of various types, including relational SQL Server data and a variety of SaaS sources. The platform includes a pipeline that enables Harmoney to ingest 50 times more data than before. Daily loads are 40 times faster and include a variety of structured and semistructured data sources to feed machine learning models. As new data comes in, it automatically runs through the model to make decisions about who to lend money to, in what amount, and at what interest rates, based on perceived credit risks.

Continually ingesting new data allows Harmoney to refresh its sales funnel every 15 minutes and to monitor the funnel via live dashboards, removing friction points for borrowers by offering prompt, accurate loan decisions. Business users now have fast, native access to data ten times faster for business intelligence, predictive analytics, and machine learning activities.

IN THIS CHAPTER

- » Balancing agility and governance
- » Establishing roles and responsibilities
- » Getting serious about data governance
- » Enforcing continuous delivery processes

Chapter 4
Establishing Your Data Engineering Foundation

Establishing a healthy and productive data engineering practice requires balancing agility with governance. You need comprehensive controls to ensure your data is clean, accurate, and up to date. However, you don't want to stymie the user community by imposing data governance procedures that are too onerous or obtrusive. Ultimately, you want an agile environment that is broadly accessible and easy to use.

This chapter describes how you can bring together a broad network of stakeholders, from highly skilled engineers and data scientists to casual users who simply want to explore, enhance, modify, and cleanse their data with self-service tools. It explains how you can enforce good governance to provide a safe environment that allows your user community to be creative, while also ensuring data is secure, consistent, and compliant with data privacy regulations.

Closing the Gap Between Governance and Agility

Many people are involved in the cycle of preparing, managing, and analyzing data. To orchestrate these efforts, you must create a cohesive environment that accommodates multiple skill sets.

Good data engineering requires a product mindset, considering data as the product, as much as it requires a specific set of tools and technologies. In recent years, principles from the DevOps world, initially created to encourage agile software development, have been applied to data modeling and design. A newer term, *DataOps,* is used to describe the practices that automate the data engineering cycle.

Data and analytics requirements change all the time, and you need *managed* self-service procedures, backed by continuous DataOps delivery methods, to keep accurate and governed data moving through the pipeline.

WHAT IS DATAOPS?

DataOps, short for *data operations,* brings together data engineers, data scientists, business analysts, and other data stakeholders to apply agile best practices to the data lifecycle, from data preparation to reporting with data. As shown in the following figure, DataOps automates critical data engineering activities and orchestrates hand-offs throughout the data management cycle, from plan, develop, build, manage, and test to release, deploy, operate, and monitor.

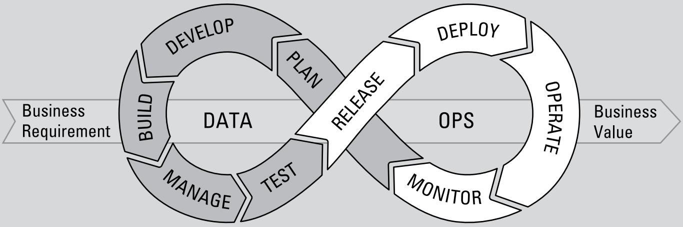

Figure source: Datalytyx DataOps

DataOps takes its cues from agile programming methods to ensure the delivery of data via a continuous plan, develop, build, manage, test, release, deploy, operate, and monitor loop. DataOps practices include tools, processes, and frameworks that recognize the interconnected nature of both business and IT personnel in these data-driven endeavors.

Good DataOps procedures enable businesses to ensure data quality, properly manage versions of data, enforce data privacy regulations, and keep data applications moving through a continuous cycle of development, integration, testing, and production. For example, as business analysts update their queries, worksheets, and schemas, data engineers must track these changes, ensure proper version control, and make sure these analytic assets still work properly and fulfill their original purposes in an automated process.

DataOps needs to start with data governance as the foundation. Consider these aspects as part of your data governance practice:

- **Lineage tracing:** Good data governance involves tracing the lineage of your data. The *lineage* of data includes its origin, where it is used, who has access to it, and what changes have been made to it over time. Tracking your data's lineage allows you to verify where data comes from and to trace errors back to the root cause in a database or analytics process. It not only helps you with compliance and auditing but also ensures you can identify and debug errors. Many internal requirements also exist for tracking data and knowing its lineage. For example, data scientists may be asked to explain why their machine learning models make certain recommendations. Being able to trace the complete lineage of the data is essential.
- **Data quality:** Data applications succeed or fail based on the confidence users have in the data. Business professionals need to know whether the data in reports, dashboards, and predictive models is correct, because the consequences of bad data are dire. Bad data can lead to faulty business decisions that result in missed opportunities, lost revenue, and escalating costs. In the case of mishandled consumer data, there can also be hefty fines.

  That's why data quality initiatives frequently begin at the top: Chief data officers (CDOs) often insist on a data governance framework that serves as a backbone for data quality, in conjunction with data security and change control procedures.

- **Data catalog capabilities:** Data catalog capabilities help organize the information within your storage. A *data catalog* is a collection of metadata, combined with data management and search tools, that helps analysts and other data users find the data they need, serves as an inventory of available data, and provides information that helps organizations evaluate the relevance of that data for intended uses.
- **Data access:** Data access rules must be established to determine who can see, work with, and change the data. Pay special attention to personally identifiable information (PII), financial data, and other sensitive information, which may need to be masked or tokenized to uphold data privacy regulations. Some cloud data platforms can apply masks and tokens to the data automatically, as well as unmask the data for authorized users.
- **Change management:** Change management utilities keep track of who accesses and changes databases and data pipelines. They track when changes were made, who made them, and which applications those changes affect. These tools help you safely manage the environment, audit usage, and trace data back to its source, reducing the chance of unauthorized alterations and errors.

Creating Your DataOps Team

Data engineering requires a coordinated effort by business and technical team members to produce valid, trustworthy data in usable and reusable forms. The following sections describe the typical roles and responsibilities of data engineers and other members of a DataOps team.

Data engineers

Data engineers make an organization's data "production ready." They create and manage data pipelines to serve different business use cases. They need to understand how to handle various data formats, how to scale data systems, and how to provide practices to enforce data quality, governance, and security. They monitor changes and enforce version control for all data objects, as well as for the data pipelines that act on them.

Depending on the data applications the business needs, the roles of data engineers are evolving and may also require expertise across system architecture, database design, data modeling, batch and streaming processing, ETL design, and managing infrastructure technologies. Although these duties vary widely, there are a few constants. Data engineers need a foundational knowledge of what modern clouds and big data consist of. They should also be proficient with programming languages such as SQL, Java, Python, or Scala. It's often necessary to understand different types of ingestion — replication and streaming protocols such as Kafka and processing frameworks such as Spark and Flink, for example.

Other team members

Data engineers don't work alone. Upholding DataOps principles requires you to involve other business and technical professionals as well, including the following:

- **CDOs** oversee all data-related functions necessary to establish and maintain a complete data strategy. Data analytics, business intelligence (BI), and data management typically fall under their purview. These senior executives are responsible for the utilization and governance of data across the organization as well as for monetizing data for new business initiatives.
- **Data analysts** often sit in the lines of business as part of the CDO organization, and they are the experts on their data for BI, knowing exactly how data is going to be used, what logic needs to be applied, and what level of quality and confidence they are looking for in the data. Data engineers often work with them to get the requirements right and enable them with self-service capabilities to access, prepare, and explore the data.
- **Data architects** are responsible for the blueprint for organizational data management, often including working with data engineers to construct data storage, processing, and query engines. This includes defining data structures, identifying data sources, and choosing the right data formats to support the models and analytic environments the user

community needs. Data architects are the technology leads who set up and enforce the overall standards of a data engineering project.

- **Data stewards** help determine what data is needed and ensure the data is properly defined. They use the data governance processes to ensure organizational data and metadata (data about data) are appropriately described and used. Generally hailing from business rather than IT, these project stakeholders understand the source data and the applications it connects to. They often get involved at the outset of a project as well as during user acceptance testing, when a data application or pipeline moves from the development phase into the testing and QA phase, and then on to production. Data stewards and product owners also play a key role in ensuring data quality.

TIP

Nominate the business users, who own and manage the data, to be responsible for data quality because they are in the best position to detect inaccuracies and inconsistencies. These data stewards can also determine how often the data should be refreshed to ensure it remains relevant, as well as when it is being analyzed out of context. Data stewards should work with data engineers to establish repeatable and consistent data quality rules, processes, and procedures.

Enforcing Data Governance and Security

Agile organizations empower users to explore, prepare, and analyze data in a managed environment. Data governance and security are the foundation of these environments. Data governance and security ensure data is properly accessed and that the access methods comply with pertinent regulatory requirements. Balanced data governance depends on three essential factors:

- **Confidentiality:** Preventing unauthorized access to data
- **Integrity:** Ensuring data is not modified inappropriately or corrupted
- **Collaboration:** Allowing disparate teams to safely and securely share curated data

FIVE STEPS TO GOOD GOVERNANCE

Key steps for establishing good data governance include the following:

1. Establish a core team of stakeholders and data stewards to create a data governance framework. This begins with an audit to identify issues with current data management policies and suggest areas needing improvement.
2. Define the problems you're hoping to solve, such as better regulatory compliance, increased data security, and improved data quality. Then determine what you need to change, such as fine-tuning access rights, protecting sensitive data, or consolidating data silos.
3. Assess what tools and skills you will need to execute your data governance program. This may include people with skills in data modeling, data cataloging, data quality, and reporting.
4. Inventory your data to see what you have, how it's classified, where it resides, who can access it, and how it is used.
5. Identify capabilities and gaps. Then figure out how to fill those gaps by hiring in-house specialists or by using partner tools and services.

Regulatory compliance is another important aspect. Many companies must adhere to government or industry regulations that control the ownership and accessibility of their data.

For example, data privacy regulations such as the European Union's General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), the Health Insurance and Portability Act (HIPAA), and the Payment Card Industry Data Security Standard (PCI DSS) are all central to enforcing data access restrictions dictating who can see what data, when, and for how long. The types of information that commonly fall under these guidelines include credit card information, Social Security numbers, health data, and consumer data. Complying with these consumer protection regulations necessitates complete and reliable data *traceability*. Organizations that work with consumer data must track where the data comes from, where it is stored, who has access to it, how it's used, and whether or not it has been fully deleted when a consumer requests it be deleted.

Streamlining DataOps by Cloning Data

Legacy and traditional technologies require you to physically create development, test, and production database environments, and physically copy tables and data among them — a costly and time-consuming process.

Once these databases become too large, DataOps professionals rarely receive a copy of the production data for development and testing purposes because it requires too many hardware and software resources to replicate and copy data.

Some cloud data platforms use cloning technology to allow DataOps professionals to do this much faster without incurring extra storage costs. Cloning technology enables you to instantly, with a single command, create logical copies of a database of any size for development and testing purposes. The best cloud data platforms support data cloning that further simplifies the process of extending data from development to testing to production.

Continuous integration and continuous delivery (CI/CD) processes work well with these native cloud architectures. It's easy to apply changes to a development environment, test new features, promote those features to QA, conduct user-acceptance testing, and deploy a new version into production for the user community. Instead of physically copying terabytes or even petabytes of data, you can instantly make a logical replica of the production data for development, testing, and QA and set up as many environments as you need.

TIP

The dynamic nature of the cloud allows you to forgo most capacity planning exercises. If you need to spin up a new database environment for development or testing, you can provision it instantly with a couple of clicks. Similarly, you can clone an existing database of any size with a single command, with no need to pre-provision storage.

IN THIS CHAPTER

- **Explaining integration styles with streaming, batch data, and data replication**
- **Transforming data to address specific business needs**
- **Delivering data to business users and data professionals**
- **Understanding the value of centralizing data with a cloud data platform**

Chapter 5
Outlining Technology Requirements

As you lay out a data architecture, select data engineering technologies, and design data pipelines, the goal is to create a data environment that not only serves your organization's current needs but also positions it for the future. To ensure you are following best practices, consider these fundamental principles:

- **Simplicity:** Are you choosing tools that minimize data movement and reduce the number of systems with which you need to interact?
- **Flexibility:** Are you creating data pipelines that are flexible rather than brittle, that uphold prevailing technology standards, and that will minimize maintenance as you deploy new data sources and workloads?
- **Scale:** Are your data pipelines architected to accommodate growing workloads, more use cases, and an escalating number of users? Can you scale data engineering workloads independently, and separately from analytic workloads?

- **Build or buy:** Will you build or buy data integration tools, utilities, and infrastructure? Will you work with commercial vendors or use open source software? Have you compared the price/performance and total cost of ownership with different data systems and the talent resources required?
- **Self-service:** Are you democratizing access to data by insulating users from complex technologies? Are you encouraging autonomy among the user community?
- **Investments:** Can you leverage existing languages, processing engines, and data integration procedures to maximize investments, minimize training, and make the most of available talent?
- **Versatility:** Can your data pipeline architecture accommodate structured, semi-structured, and unstructured data based on your needs, as well as batch and real-time data?

This chapter discusses how you can uphold these principles as you ingest, replicate, transform, and deliver data throughout your business, as well as create continuous integration (CI) and continuous delivery (CD) processes for the IT pros tasked with managing the data environment.

Ingesting Streaming and Batch Data

The goal of data ingestion is to create a one-to-one copy of the source data and move it to the destination platform. The cycle begins with identifying your data sources and figuring out how to ingest those sources, whether as a continuous stream or via batch/bulk loads.

Dealing with streaming data

Streaming data includes IoT data, weblog data, and other continuous sources emitted by mechanical equipment, environmental sensors, and digital devices such as computer hardware and mobile phones.

The destination for this data could be a data lake; an alternative would be to ingest data from publishing/subscribing messaging services directly into a cloud data platform, perhaps augmented with an object storage service such as Amazon Simple Storage Service (S3), Microsoft Azure Blob, or Google Cloud Storage.

Distributed publishing/subscribing messaging services represent a popular way to send and receive streaming data. These services act as publishers and receivers to ensure that data is received by the subscribers. Examples include open source technologies such as Apache Kafka and commercial technologies such as Amazon Kinesis, Microsoft Event Hubs, and Google Cloud Pub/Sub.

Working with batch loads

Batch ingestion processes are commonly used for application data such as the online transaction processing (OLTP) data that underlies an enterprise resource planning (ERP) or customer relationship management (CRM) system. The data is ingested via bulk loads, and aided by data replication tools and extract, transform, and load (ETL) tools or extract, load, and transform (ELT) tools. Data engineers must determine what type of analytic repository they wish to load this data into and how often to refresh the data to meet the business requirements.

For example, retail point-of-sale data may only need to be updated in a data warehouse at the end of each day to accommodate daily revenue reports. Customer data in a CRM system may need to be uploaded to a call center dashboard once per hour to reflect current sales and service transactions. And electricity usage data collected from smart meters may need to be refreshed every 15 minutes to support time-of-use billing programs.

TIP

Some cloud data platforms include scalable pipeline services that can continuously ingest data without affecting the performance of other workloads. Data engineers can decide how much computing power to allot to each data ingestion process, or allow the system to scale automatically. These platforms also allow data engineers to build data pipelines with a wide choice of languages and integration tools for managing the ingestion stream.

Replicating existing databases

Data replication tools are valuable for data engineering projects that involve preexisting database sources and APIs, such as structured data in a relational database or data warehouse. Typically, you will want to load that data intact, since it has already been carefully modeled.

These tools can move the data while maintaining the same structure and values, producing a one-to-one copy of the source. First, they establish the structure in the destination location, usually a table or some other type of database object within a cloud repository. Then, they incrementally move the data. They manage the initial bulk data load to populate the destination database followed by periodic incremental loads to merge new data and keep the destination database up to date. These are generally two separate ETL processes you would have to create manually, so using a replication tool is a big timesaver.

Data replication tools commonly use Structured Query Language (SQL) MERGE statements (also called *upsert* statements) to insert new records and update existing records. They also handle deduplication operations to ensure data integrity, deal with schema changes, and monitor the entire process for errors. If you already have existing tables, structures, and relationships you want to maintain, a replication tool will maintain those artifacts and do most of the ingestion work for you.

REMEMBER

Ingestion, or loading data, can be a difficult part of data engineering, but if you do it right, you will have a one-to-one copy of your source data in a destination repository, such as a cloud data platform, data warehouse, or data lake. Ingestion doesn't have to involve any transformations. The destination data can look exactly like the source data, including the same table names, column names, and so on.

Transforming Data to Address Specific Business Needs

Although ingestion is all about getting the data in place, *transformation* involves solving particular problems with the data by transforming it to address specific business needs. The goal of data transformation is to integrate, cleanse, augment, and model the data so the business can derive more value from it. Transformation processes ensure the data is accurate, easy to consume, and governed by consistent business rules and processes. You must determine what business logic you will need to incorporate and how you want to transform or structure the data to uphold the business rules. Ask yourself what you are trying to build, and why, such as what business processes you want to model or create.

Creating data transformations has two main phases: *design* and *execute*. The following sections describe these phases.

Designing transformation logic

During the design phase of data transformation, you determine whether you will hand-code the transformations with an integrated development environment (IDE) or use a data integration tool to pull your pipeline together.

To decide which approach to take, think about what you have to start with. For example, do you have existing transformations you want to carry forward? How much of that logic can you leverage in your new pipeline, and what type of talent pool do you have to adapt the existing interfaces? How much flexibility are you looking for when it comes to customization? Wherever possible, try to build on your existing investments and skill sets. Data engineering logic, interfaces, and procedures can often be reused.

Some cloud data platforms allow you to design these jobs with SQL, others with Python, Java, or Scala. The talent and resources available to you are key decision factors in choosing your platform, but also look for extensibility of data pipelines so you can allow users to bring in their own language of choice when designing the transformation logic. This encourages more collaboration for data engineering.

Of course, as discussed in Chapter 2, design has everything to do with where you want to execute and process the data.

Executing data processing

Transformation processes require lots of compute cycles. During the processing phase, you must determine where you will run your processing engine and what performance you will require.

Do you have any performance or concurrency issues, such as limited server capacity? ETL processes are often handled by on-premises servers with limited bandwidth and CPU cycles. Modern processing engines push these resource-intensive transformation workloads to the cloud so you can run the transformation logic *after* you ingest the data. In this case, you should use an ELT process to get your data into the cloud database so you can use the boundless resources of the cloud to process and transform it quickly and efficiently.

Cloud services today provide more options for compute and storage resources. Look for a cloud data platform that allows you to isolate your transformation workloads from each other and automatically allocate appropriate compute capacity to handle the integration and transformation work without degrading the performance of your analytic workloads.

CASE STUDY

SIMPLIFYING DATA INGESTION, TRANSFORMATION, AND DELIVERY

Panoramic is a world leader in providing analytics to communications firms and media service providers. It helps marketers use consumer data to hone their creative strategies. Panoramic's data scientists and marketing analysts build customized data platforms that enable marketers to visualize consumer data for analysis, benchmarking, internal collaboration, and more. The company has built its business around ingesting and modeling marketing data as a foundation for generating meaningful insights.

Panoramic constantly collects, aggregates, and manages semi-structured data from marketing and advertising platforms such as Facebook, Google Ads, Twitter, Amazon, and Bing. The data engineering process is complicated by the fact that the APIs and data schemas from these platforms are constantly changing. These shifting requirements forced Panoramic's data engineers to spend many hours each week adjusting the schemas, modifying APIs, and loading data into the marketing intelligence platform.

To simplify data ingestion and transformation activities, Panoramic adopted a cloud data platform that includes a native data-ingestion process for storing and managing semi-structured JSON data. The new platform gives Panoramic an easier way to handle the flexible schemas associated with JSON documents, optimizing data ingestion, simplifying data sharing, and facilitating workflow automation. The platform saves many hours and thousands of dollars formerly spent managing APIs, data schema, and ETL procedures.

Today, Panoramic's data engineers can focus on delivering value, rather than on working with tedious data interfaces. And because the platform allows for secure data sharing without copying or duplicating data, it's easy to extend these robust data sets to customers.

TIP

Whenever possible, use ELT (instead of ETL) processes to push resource-hungry transformation work to your destination platform. This provides better performance, especially if that destination is a scalable cloud service. Ideally, you should process the data where that data resides rather than moving it to an independent server or storage mechanism.

Delivering Data to Consumers and Data Professionals

Once data is transformed, data engineers need to make sure it is delivered appropriately to users within and outside of your organization. Your data management environment should allow you to seamlessly and securely share data internally among authorized users, departments, and subsidiaries, as well as externally with partners, suppliers, vendors, and even customers.

To do this efficiently, your data management environment should allow you to extend *live access* to any subset of your data to any number of data consumers, inside and outside of your organization. You'll want a data environment in which all database objects are centrally maintained, governed, and secured.

A cloud data platform allows you to maintain a single source of truth that facilitates data sharing, monetization, and exchange. Because data is shared rather than copied, no additional storage is required.

REMEMBER

To make your data accessible to consumers, both internally and externally, you need a modern data sharing architecture that allows you to extend secure access to data to authorized users without moving or copying the data.

Using CI/CD Tools

DataOps has become critical for delivering data in a timely and productive way. As discussed in Chapter 4, data engineers need to be able to set up a continuous plan, develop, build, manage, test, release, deploy, operate, and monitor loop to accomplish this.

CONTINUOUS IMPROVEMENT CYCLE

To make sure all your software tools and utilities work with your automated CI/CD cycle, verify that these requirements have been met:

- Can you build data pipelines with leading data integration tools?
- Can you easily seed preproduction environments with production data?
- Can you instantly create multiple isolated environments to do your validations?
- Can you scale the data environment to run validation jobs quickly and cost-effectively?
- Can you clone data and immediately spin up compute clusters for development and testing purposes?
- Can you control your schema with change management tools and keep track of versions for development, testing, and auditing purposes?
- Can you automate CI/CD pipelines with your preferred software automation tools?
- Can you restore data easily by rolling back to previous versions and points in time?

Organizations that follow IT best practices manage their data environments using CI/CD tools with modern cloud solutions. Your data engineering tech stack should support the use of DataOps tools and processes. For example, you should be able to incorporate your chosen data replication tools, ETL tools, data transformation tools, data change management tools, and data management utilities into a continuous delivery cycle.

Meeting All Your Needs with a Cloud Data Platform

The ground rules are clear: Today's organizations want to cost-effectively load, transform, integrate, and analyze unlimited amounts of structured and semi-structured data. They want to collect, store, and analyze their data in one place, so they can easily obtain all types of insights from their data. And they want to simplify and democratize the exploration of that data, automate routine data management activities, and support a broad range of data and analytics workloads.

A properly architected cloud data platform brings this all together with a highly integrated set of services that streamline how data is used. It enables you to consolidate diverse analytic activities, orchestrate the secure sharing and exchange of data, and populate modern business applications with data (see Figure 5-1).

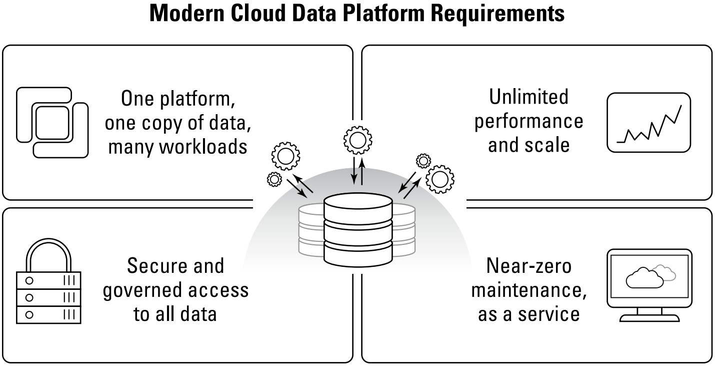
Modern Cloud Data Platform Requirements
One platform, one copy of data, many workloads
Unlimited performance and scale
Secure and governed access to all data
Near-zero maintenance, as a service

FIGURE 5-1: The fundamental elements of a modern cloud data platform.

By creating a single place for all types of data and all types of data workloads, a cloud data platform can dramatically simplify your infrastructure, without incurring the costs inherent in traditional architectures. For example, by centralizing data, you reduce the number of stages the data needs to move through before it becomes actionable, which eliminates the need for complex data pipeline tools. By reducing the wait time for data, you allow users to obtain the insights they need, when they need them, so they can immediately spot business opportunities and address pressing issues. One unified platform can handle everything you used to do in a data warehouse, data lake, and multiple data marts.

Modern cloud data platforms include critical capabilities to set the foundation for reliable and efficient data pipelines for moving all your data to the cloud. From there, the data undergoes various transformations until it is usable for analytics, data science, data sharing, and other business initiatives.

Modern data pipelines have five unique characteristics:

- Continuous and extensible data processing
- The elasticity and agility of the cloud to expand storage capacity
- Isolated and independent resources for data processing
- Democratized data access and self-service management
- Support for continuous integration/continuous delivery processes

A cloud data platform makes it much easier to provide convenient access to all your data, and it improves the speed with which you can analyze and share data across your extended enterprise.

Data engineering is a broad practice and involves many people, processes, and technologies. You may not find all the capabilities you need in one single solution, but you'll want to start with a cloud data platform that can address the majority of your requirements.

IN THIS CHAPTER

- Working with existing data sets, applications, and resources
- Leveraging the cloud to simplify your operation
- Enlisting qualified team members
- Enforcing data governance
- Making wise architectural choices
- Encouraging self-sufficiency

Chapter 6
Six Steps to Building a Modern Data Engineering Practice

The previous chapters outline the various technology and organizational decisions you need to take into account in your data engineering endeavors. This chapter offers six guidelines for putting those decisions into practice.

1. Start Small, Think Big

Take a close look at the current state of data engineering within your organization. Which part of your current architecture needs to be modified or enhanced to support more robust data pipelines? What analytics initiative should you start with?

Identify a manageable use case — ideally one that will have an immediate impact on your business. Urgent projects often center around operations improvement, such as identifying inaccurate data or fixing an inefficient data pipeline that takes too long to run.

Longer-term initiatives focus on opening up new revenue opportunities. In both cases, ask how you can extend your current technology assets. What tools have you invested in? Where can you benefit the most by replacing legacy tools with modern technology? Start with one project, and move on to the next. Gradually, think about how you can establish an extensible architecture that leverages the data, tools, and capabilities you have in place while incorporating the modern tools, processes, and procedures described in this book.

2. Simplify Your Architecture

Data is fundamental to the workings of the enterprise. But if you want to push your data to consumers fast with limited resources, you need to simplify your data architecture. Get out of the business of managing infrastructure. Help the business leaders at your organization understand a basic fact: The quickest way to deliver analytic value is to stop managing hardware and software.

Managed cloud services allow businesses large and small to dynamically expand and contract their information systems instantly and near-infinitely, automatically or on the fly. What's the upshot? IT professionals have fewer systems to host and fewer "knobs" to turn, meaning less manual tuning and administration. Look for technology that is hosted by a cloud vendor but that is flexible enough to meet your needs.

3. Enlist Help from Stakeholders

As discussed in Chapter 4, data engineering is a team sport. How do you enlist the right team members? Follow the data. Appeal to the users, managers, and departments that have the most to gain. Build allegiances, engage analysts, and try to acquire high-level support from executives and directors.

As your data engineering efforts expand, think about how you can scale the team. Leverage existing data engineering resources as well as skilled IT personnel who may be eager to move into new roles. Review the requirements discussed in Chapter 5, and build the skill sets you need to support data ingestion, preparation, transformation, exploration, and delivery, ideally based on DataOps procedures and continuous integration/continuous delivery (CI/CD) methods.

4. Don’t Make Data Governance an Afterthought

Once you have alignment with business and IT, identify product owners to oversee data quality. Remember the fundamentals of data governance, data security, curation, lineage, and other data management practices outlined in Chapter 4. Does your organization already have a DevOps strategy? Find out who spearheads this effort and if they are familiar with the principles of DataOps as well. DataOps practices help set good data governance foundations so you can empower users to self-serve as they prepare, explore, analyze, and model their data, using fresh data in good quality.

As you approach each data engineering project, pay close attention to the following:

- What industry or data privacy regulations must you observe?
- Who are the data stewards, and how will they set up processes and workflows to enforce data governance and integrity?
- Can you enforce role-based access and fine-grained privacy control?
- Do you have data lineage capabilities for tracing the journey data follows from original source to final destination?
- Can you organize data so discovery is enabled for your users?

5. Maximize Efficiency with a Cloud Data Platform

As discussed in Chapter 5, it’s often better to process data once it reaches its destination, especially if that destination is a scalable cloud service. Either way, transforming data requires lots of compute resources. Whenever possible, leverage the modern data processing capabilities of the cloud.

The best cloud data platforms include scalable pipeline services that can ingest streaming and batch data. They enable a wide variety of concurrent workloads, including data warehouses, data lakes, data pipelines, and data exchanges, as well as facilitating business intelligence, data science, and analytics applications.

Consolidating data into a single source of truth, whether it be in a single location or across multiple repositories, makes it easier for data consumers. Once data is in one place, it's easier to access, analyze, and share with other constituents in the cloud ecosystem. It also allows IT professionals to shift their focus from managing infrastructure to easily managing data as a single source of truth.

6. Look to the Future

As you select data engineering technologies, design data pipelines, and establish new data architectures, think about how you can serve your organization's current needs while positioning it for what lies ahead, including more advanced data science initiatives such as machine learning and deep learning.

In the modern enterprise, data is everywhere, and everybody is a decision maker. How can you extend data-driven decision-making capabilities to executives, managers, and individual contributors across your organization? These knowledge workers shouldn't have to look for data. It should be infused naturally into the apps they use every day, and presented within the context of their day-to-day activities.

The goal of the data engineer is to move beyond serving a small group of data scientists and analysts, and to empower the other 90 percent of workers who depend on data to do their jobs. How do you get there? Adopt a product mindset. Strive to impact corporate goals connected with generating revenue, maximizing efficiency, and helping your people discover new opportunities for your organization. Think about how you can share, monetize, and exchange corporate data to create new business value.

Ensure all your data is reliable, high-quality, and delivered rapidly for up-to-the-minute analytics

Getting all the insights from all your data to all your business users starts with modern data engineering. It is the first and most essential step of delivering analytics-ready data to democratize business intelligence and advanced analytics across your organization. It is also the competitive edge you need to best serve your customers, reduce costs, and lead your industry. This book reveals how your organization can build a modern data engineering practice to produce fast, reliable, and quality data for all your business units, and to securely share data within your organization and with your ecosystem of customers and business partners.

Inside…

- The fundamentals of data engineering
- Collecting, transforming, and delivering data
- Building efficient, modern data pipelines
- Determining roles and responsibilities
- Defining your technology requirements
- Steps for building a modern data engineering practice
- Real-world data engineering case studies

snowflake™

David Baum is a freelance business writer specializing in science and technology.

Go to Dummies.com™
for videos, step-by-step photos, how-to articles, or to shop!

ISBN: 978-1-119-75456-5
Not For Resale

9 781119 754565

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT