The Essential Guide to Data Integration

How to thrive in an age of infinite data

By Charles Wang

Discussed in this guide:

- How data integration fuels analytics
- The evolution from ETL to ELT to automated data integration
- The benefits of automated data integration
- How to evaluate data integration providers

Fivetran

fivetran.com →

Table of Contents

About This Book

Objectives

This book will demonstrate the usefulness of data integration to your organization, explain and evaluate the various approaches to data integration currently available, and show you how to implement the technology. You don't need to read it front to back, although it is organized to be digested easily that way.

Data integration consists of the processes used to manage and centralize flows of data from various sources, in order to use that data to guide decision-making. The practical interpretation of data to guide decisions is often referred to as **analytics**. As we will see, the quality of your analytics program is closely tied to the quality of your data integration technology.

Data integration allows your organization to maintain all of its data in a single environment, so your team has a comprehensive view of business operations and customer interactions. Centralizing data and making it accessible promote widespread data literacy, enabling organizations to spot hidden opportunities, improve performance and spur innovation.

In this guide, we will discuss:

- What data integration is and why it's important
- The traditional approach to data integration, called extract-transform-load (ETL)
- The newer approach enabled by the cloud, called extract-load-transform (ELT)
- The value of automating the data integration process
- How to evaluate and adopt data integration tools

Intended Audience

To make the most of this guide, you should have some familiarity with data engineering, data warehousing, analytics, business intelligence, data visualization and related concepts. We assume that you belong to an organization that uses operational systems, applications and other tools that produce digital data, and that some of your operations are already based in the cloud. We also imagine that your role — analyst, data engineer, data scientist or manager of the preceding titles — puts you in a position to influence or determine the tools your company uses.

Icons Used in This Guide

As you read, you will encounter icons for tips, warnings, important points to remember and case studies.

TIP: *Practical advice regarding data integration and analytics*

CAUTION: *Instances of data or technological malpractice*

REMEMBER: *Important points worth remembering*

CASE STUDY: *Real-world data integration success stories*

Beyond the Guide

If you find this guide useful and want to learn more, take a look at *fivetran.com/blog*, where we publish new content about data engineering and analytics. Another good resource is our documentation, *fivetran.com/docs*, which offers a detailed look at how automated data integration works for specific data sources and destinations.

Chapter 1: Data Integration and Analytics

IN THIS CHAPTER:

- Historical overview of analytics
- The value analytics offers you and your organization
- A major obstacle to analytics: data integration

History of Analytics

Analytics long predates modern data collection. Florence Nightingale used coxcomb diagrams to identify and reduce the causes of hospital mortality during the Crimean War (Figure 1.0). William Sealy Gosset, chief brewer at Guinness, developed Student's t–test to ensure beer quality. People have long extracted valuable, practical stories and lessons from numbers.

In the years since, statistics has continued to mature as a science, as have the tools and methods used to analyze data. The growth of modern computing and the internet, in particular, has enabled the collection and analysis of data at vastly larger scales than was possible using pen, paper and tabulating machines.

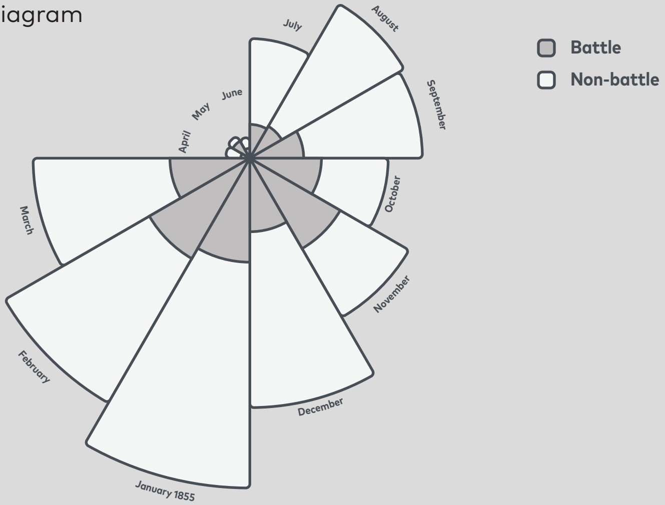
Goals of Analytics

Analytics offers competitive value in several ways. It can be used to improve customer acquisition, retention and loyalty; identify new product opportunities; and enhance existing opportunities. By improving organizational decision-making, analytics can deliver many times its cost in ROI.

Broadly speaking, you can use analytics in the following ways:

1. **Ad hoc reporting**. Key stakeholders and decision-makers will sometimes need very specific questions answered on a one-time or occasional basis.
2. **Business intelligence**. Often used interchangeably with "analytics," business intelligence (BI) refers to the use of visualizations and data models to identify opportunities and guide business decisions and strategies. This usually comes in the form of regular, consistent reports and up-to-date dashboards.
3. **Data as a product**. Data your organization collects or produces can be made available to third parties in the form of embedded dashboards, data streams, recommendations and other data products.
4. **Artificial intelligence/machine learning**. The pinnacle of analytics is building products and systems that use predictive modeling to automate important decisions and processes.

Figure 1.0
Coxcomb Diagram



Source: "Notes on Matters Affecting the Health, Efficiency and Hospital Administration of the British Army," by Florence Nightingale. London: Harrison & Sons, 1858.

At the organizational level, analytics can also help you pursue the following:

1. **Democratizing access to data/data literacy**. As more and more employees

use data to make decisions, your organization will react more intelligently to changing circumstances. With the right BI tools, even non-technical team members can make decisions based on data. Of course, this both requires and allows you to afford your team a great deal of trust and latitude.

2. **Enhancing your products and services**. Insights gained from analytics will help you improve your offerings and provide additional transparency and reporting options to your customers.
3. **Keeping your organization competitive**. Data literacy allows you to make the most of limited resources and uncover opportunities that would otherwise be invisible.

Knowledge is power, and it is always advantageous to know more than the competition.

TIP: *It can be useful to think of all data-related activities as stages in a hierarchy of needs, in which satisfying foundational needs enables the pursuit of higher needs (Figure 1.1).*

The most basic need is the collection and storage of raw data, i.e., data integration. Once that need is satisfied, the intermediate needs of analytics and predictive modeling become easier to satisfy. This allows your organization to create a data-driven culture, in which every employee has access to the data they need to make better-informed decisions.

At the very top of the hierarchy, data is used to train machine learning models and develop artificial intelligence, enabling the automation of workflows and decision-making within your organization and the creation of "smart" consumer-facing products.

A Major Obstacle to Analytics: Data Integration

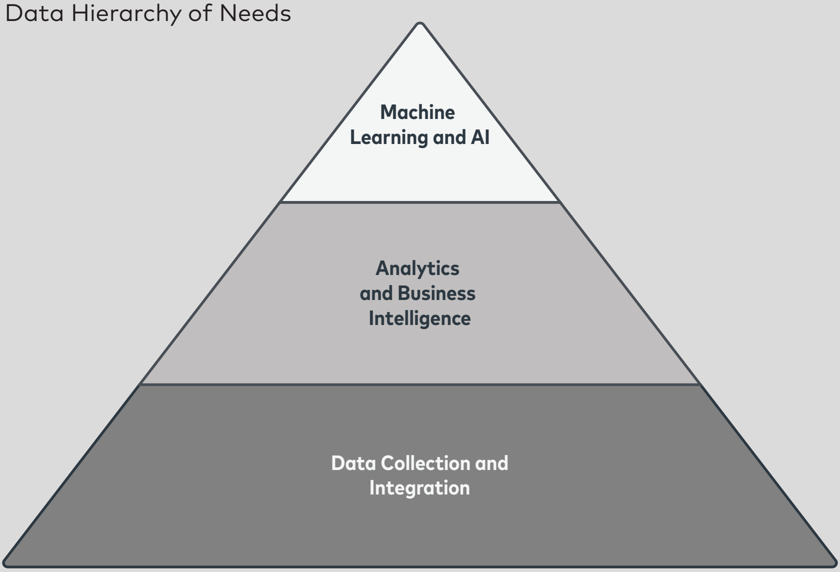
A central data repository of record offers your organization the following benefits:

1. You gain a big-picture view of your organization's operations and see how the parts work together, instead of viewing siloed, isolated representations.
2. You can match records and track the same entities (customer, partner, etc.) across different stages of their life cycles.

3. You can perform analytics in an environment separated from operational systems, preventing your queries from interfering with your operations.

4. You exercise granular control over access and permissions, ensuring that your team gets the information they need to perform their jobs without compromising sensitive systems.

Figure 1.1
Data Hierarchy of Needs


Machine Learning and AI
Analytics and Business Intelligence
Data Collection and Integration

Creating this central data repository can be a Herculean task. Every data source requires separate procedures and tools to ingest, clean and model its data. This challenge has been amplified by the recent proliferation of cloud-based applications and services. The appearance of web-enabled devices and sensors (i.e., the Internet of Things), has likewise contributed to an explosion of data (Figure 1.2). Since 2013, it has been a truism that 90% of the world's data was created in the last two years.[1]

Where Does Data Come From?

Data can originate from:

1. **Sensor inputs**, such as scans at a checkout line

[1] sciencedaily.com/releases/2013/05/130522085217.htm

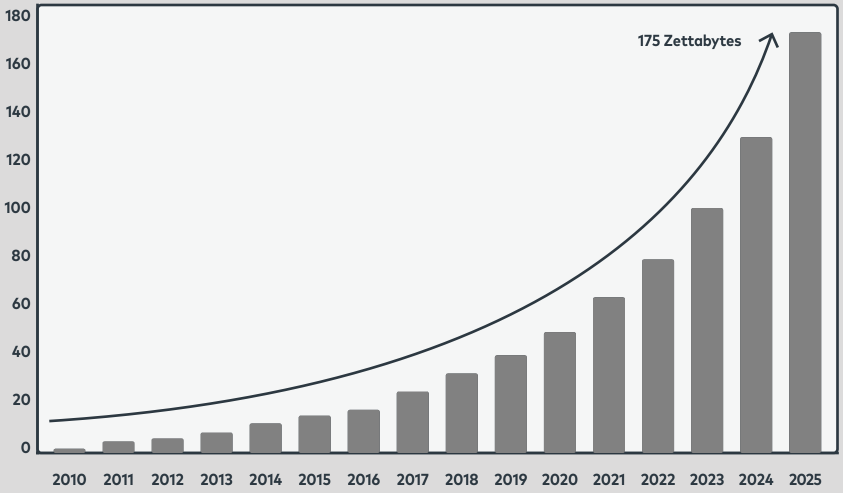
2. **Manual data entry**, such as forms collected by the Census Bureau
3. **Digital documents and content**, such as social media posts
4. **Digital activity** recorded by software triggers, such as clicks on a website or app

Data from the above sources is typically stored in cloud-based digital files and operational databases, and then exposed to an end user in the form of:

1. API feeds
2. Files
3. Database logs and query results
4. Event tracking

API feeds allow applications to communicate with each other, often by exchanging data in formats like JSON or XML. Most organizations use a wide range of applications to handle operations such as customer relationship management, billing, customer service and more. APIs enable data ingestion and interoperability between software applications.

Figure 1.2
Annual Size of the Global Datasphere



Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018

Data **files** such as CSV, XLSX and TSV can originate from multiple activities across an organization, from manual data collection to ad hoc calculations.

Database logs and query results are generated by operational databases updated in real time. They support day-to-day interactions for everything from sensors to software. An ecommerce website, for instance, might use an operational database to record purchases, listings and customer profiles.

Event tracking occurs via user-triggered code snippets embedded in web pages and applications. A basic tool might record button clicks in an app; a more advanced one might track cursor position. An even more sophisticated tool might use a laptop camera to track a user's eye movements. Event tracking data generates a granular record of how users interact with a website or application, and is particularly useful for UI/UX research. One of the most common forms is the **webhook**, which is embedded in web apps and sent over HTML rather than formatted in XML or JSON.

As you can imagine, the sheer diversity of data sources and formats creates significant challenges for data engineers trying to integrate and normalize data streams.

TIP: *You may occasionally hear the phrase "data integrity" with regard to various kinds of data operations, including data integration. Data integrity refers to the completeness, accuracy and consistency of data through all stages of its use. Violations of data integrity include mistyped or incorrectly formatted data, duplications, omissions, and incorrect relations between tables.*

SaaS Data: A Growing Challenge and Opportunity

In the cloud era, SaaS applications have become one of the predominant sources of business data. SaaS apps span a huge range of operations and industries: marketing, payment processing, customer relationship management, ecommerce, engineering project management and many more. They provide sophisticated services and operations, obviating the need to construct tools in-house or rely on massive outlays of labor to perform the same tasks manually.

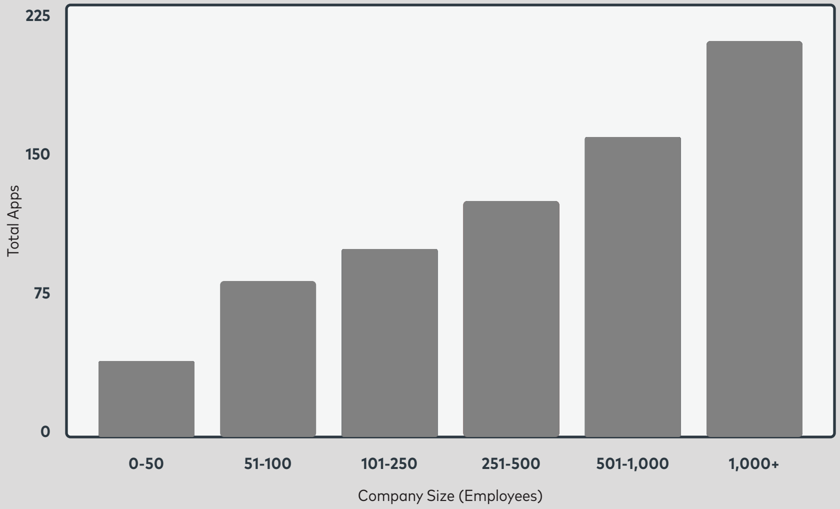
SaaS applications commonly record actions by users, offering organizations highly granular pictures of their operations, from which they can deduce patterns and causal relationships. Generally, the more facets of your business you can quantify and analyze, the more competitive you are.

However, voluminous data poses an overwhelming data integration challenge. A typical company now uses more than 100 apps (Figure 1.3). At that scale, manual data integration is virtually impossible. As we will see, many organizations still write bespoke software and create custom infrastructure to integrate data, but that approach becomes untenable when data comes from dozens of sources that generate a continuous, high-volume data throughput.

Even at smaller scales, the workload imposed by building and maintaining complex data pipeline software can hobble analytics efforts. Heavy time commitments divert analysts, data scientists and engineers from other activities.

Luckily, cloud technology offers a solution to this challenge. Modern data pipeline tools, data warehouses and business intelligence platforms are cloud-based applications in their own right, and they have proliferated alongside cloud technology. They effectively eliminate the need to manually develop customized, in-house tools and solutions for data integration and analytics.

Figure 1.3
Number of Apps per Company



Source: Blissfully's 2019 Annual SaaS Trends Report

REMEMBER: *Analytics capabilities empower companies to do their best work, but to make your analytics performant and comprehensive, your data must be accessible in a central environment. A central repository of data allows your organization to:*

- *Gain a complete, big-picture view of your organization's operations*
- *Match records and track the same entities across different data sources*
- *Separate analytics from operational systems*
- *Control access and permissions*

Data integration, the process of centralizing data and making it accessible, is both extremely important and extremely difficult to perform well. Be sure to take the challenge seriously.

Chapter 2: Approaches to Data Integration

IN THIS CHAPTER:

- What is data integration and how is it performed?
- The traditional approach (ETL) vs. the modern approach (ELT)
- How automated ELT solves data integration

The Basic Data Integration Process

Data integration consists of the following steps:

1. Data is gathered from sensor feeds, manual data entry or software, and stored in files or databases.
2. Data is extracted from files, databases and API endpoints and centralized in a data warehouse.
3. Data is cleansed and modeled to meet the analytics needs of various business units.
4. Data is used to power products or generate business intelligence.

Data integration can be performed in a manual, ad hoc manner or programmatically, using software. The ad hoc approach is unreplicable and non-scalable, while the programmatic approach requires a **data stack** containing a distinct set of complementary tools. In the rest of this chapter we will discuss these concepts, as well as the history and future of data integration and data stacks.

Non-Scalable Approaches to Data Integration

Many organizations rely on a manual, ad hoc approach to data integration — in fact, 62% use spreadsheets like Excel and Google Sheets to stitch together elements from data files and visualize data.[2] This involves downloading files, manually altering or cleaning values, producing intermediate files, and similar actions. Ad hoc data integration has a host of drawbacks; specifically, it is:

- Suitable only for very small volumes of data
- Slow
- Prone to human error
- Insufficiently secure for sensitive information
- Often unreproducible

A more sustainable approach is to maintain the silos between separate data sources while bridging the gaps between them with "federated" queries, which directly query multiple source systems and merge data on the fly. Organizations may do this with SQL query engines like Presto. The disadvantage of this federated approach is that it involves many moving parts, and its performance degrades at large scales of data.

The reality is that a scalable, sustainable approach to analytics requires a systematic, replicable approach to data integration — a data stack.

Data Integration With a Data Stack

A **data stack** consists of tools and technologies that collectively integrate and analyze data from a variety of sources. The components of a data stack include:

1. **Data sources**:

    a. Applications
    b. Databases
    c. Files
    d. Digital events

2. **Data pipeline** and **data connectors**. Software used to extract data from a data source and load it into a data warehouse. This accounts for the bulk of data integration.

[2] zdnet.com/article/spreadsheets-still-dominate-business-analytics/

3. **Data warehouse** and/or **data lake**. A data repository of record designed to permanently accommodate large amounts of data. Data warehouses are almost always column-based and contain data in a relational structure, while data lakes are object stores that can contain both structured data and raw, unstructured data.
4. **Data modeling** and/or **transformations**. Oftentimes, it is necessary to prepare your data by applying custom business logic, such as changing column names or conducting aggregations, to get it ready for your specific analytics use case.
5. **Business intelligence tool**. Software meant for summarizing, visualizing and modeling data in order to guide business decisions.

CAUTION: *Data lakes and data warehouses have traditionally stored different types of data in order to facilitate different use cases. Data lakes typically include raw, unstructured data and are therefore murky. This unstructured data is not cleansed, normalized or transformed before it lands in the destination system, leaving data scientists with the burden of wrangling the data into a usable state. Data warehouses contain tables organized by columns and are generally based on traditional relational databases that can be queried using SQL.*

More recently, data lakes and data warehouses have begun to evolve in a convergent manner. Data lakes have begun incorporating ACID (atomicity, consistency, isolation, durability) transactions and schema enforcement as features to make data less "murky." Likewise, data warehouses, which can already perform ACID transactions, have begun to support data science tools and languages usually associated with data lakes such as Apache Spark and Python.

Convergent evolution aside, data lakes are better suited for use cases in which support for machine learning, artificial intelligence and an open ecosystem of data science tools is more important than accessibility. The end users of data lakes are typically highly skilled data scientists with experience using Spark, Python, Pandas and similar tools to wrangle a variety of data types at scale. Data warehouses are better suited for operational analytics and business intelligence use cases, in which end users rely primarily on SQL and BI dashboards.

How Data Moves Through the Stack

The most basic unit in a data pipeline is a piece of software called a **data**

connector. A data pipeline may contain one or several connectors, each of which extracts data from a source — an app, event tracker, file or database — and usually applies normalization and light cleaning.

The data is then routed to a **data warehouse**. **Transformations** can either be performed before the data arrives in a data warehouse, or within the data warehouse after the data arrives. This is what separates ETL from ELT — but more on that later. In both cases, transformations may be orchestrated — that is, arranged in a sequence with automated logic to coordinate sequencing, timing and errors. Ideally, data warehouses serve as a repository of record for the entire organization. Any kind of relational database can be used in the data warehouse role, but data warehouses are typically columnar in nature, as opposed to transactional or production databases, which tend to be row-oriented and therefore less efficient for analytics queries.

Finally, the data is analyzed with the help of a **business intelligence tool**. Business intelligence tools commonly display trends, proportions and other findings on dashboards and in periodic reports.

The individual components of a data stack can be hosted on-premise or in the cloud. Traditionally, organizations have used on-premise data stacks. Although the cloud has grown in prominence, some organizations remain on-premise in the interests of regulatory compliance or highly specific performance needs. They may build major components of their data infrastructure in-house to avoid external dependencies or vendor lock-in.

TIP: *A technical discussion of the difference between row-oriented and column-oriented databases is beyond the scope of this guide, but here's a quick primer. Row-oriented databases — also called online transaction processing, or OLTP, databases — are typically used in production to handle individual transactions. Column-oriented databases are typically better at handling the columnar operations used in analytics (MIN, MAX, SUM, COUNT, AVG). Familiarity may tempt you to simply make a copy of your production database to use for analytics. Don't do it! Always use a column-oriented database or data warehouse for analytics. It will be more efficient and save you a great deal of time.*

Challenges a Data Stack Can Solve

As it funnels data from connectors to data warehouses, a data stack must ensure that data is centralized in a single environment and that it remains as current and true to its source as possible. The process should be conducted continuously, with minimal human intervention.

Fragmentation

Data often arrives from apps, tools and databases in a fragmented state. There are two types of fragmentation. The first occurs because API endpoints and operational databases are not designed for analytics queries, meaning the data they generate often lacks important context and isn't organized to facilitate analytics. Extensive data modeling is often required to make sense of the data.

The second occurs because most apps, tools and databases are not specifically designed for interoperability with data from other systems. The burden of establishing the necessary context by joining records across multiple sources can result in very long reporting turnaround times.

Thus the phenomenon of "dark data," which occurs when a large fraction of the information assets collected by a typical company goes unused. Centralizing data in a single environment enables faster reporting and allows organizations to join records and construct cohesive narratives about their operations and customers. Property broker Zoopla, for example, combined ERP and CRM data to produce a weekly dashboard featuring more than 40 separate KPIs.

CASE STUDY: DiscoverOrg Stopped Using OLTP for Analytics

DiscoverOrg is a B2B lead-generation platform that profiles individuals and companies in order to enable smarter sales and marketing efforts. Before building a data stack, DiscoverOrg drew analytics data from a copy of its OLTP production database, and excluded data from any third-party applications. Queries could take up to 36 hours, or crash the system.

After adopting an automated data integration tool, DiscoverOrg was able to combine its production data with data from third-party sources into a data warehouse, spare the work of two or three data engineers, generate reports in a matter of minutes instead of days, and develop a lead-routing algorithm to acquire contracts with an 80-90% higher average value.

More recently, DiscoverOrg has begun embedding analytics dashboards within its platform for the benefit of its customers.[3]

Accuracy

There are two senses in which data can be inaccurate. One is faulty measurement or recording, especially if the data was entered by hand or transcribed from non-digital

[3] Read the full case study at fivetran.com/blog/case-study-discoverorg

media. Surveys and forms are bound to produce misspellings, transposed characters and other clerical errors. A second and more systematic source of error comes from calculations or transformations performed on raw data. There are many ways to massage a data set, and every calculation takes you one step further from the original values. Thus different people and teams within an organization can arrive at radically different versions of the truth.

Stale Data

External conditions change quickly. Spend weeks or months assembling a report, and you may make seriously misguided decisions because you're working with obsolete data. People familiar with decision models like PDCA (plan-do-check-act) or OODA (observe-orient-decide-act) understand the importance of making informed decisions more quickly than the competition. These kinds of models apply to all competitive, dynamic environments, including warfare, gaming, athletics and, of course, business.

CASE STUDY: Zoopla Uses a Data Pipeline to Unify Data Integration Efforts

Zoopla is an online property market that enables users to buy, sell or rent residential or commercial property in the UK.

Before Zoopla adopted a modern data stack, its analytics efforts were scattered. Analysts and engineers built a variety of custom scripts on an ad hoc basis to extract and analyze its data. These scripts were not documented and were often written in different languages. Analysts also used native data connectors in Zoopla's BI tool to perform federated queries.

The BI team at Zoopla recognized that the old arrangement was unsustainable as the company continued to grow and add data sources, and as it strove to quantify its progress. After adopting a modern data stack, Zoopla was able to combine data from its ERP and CRM software to produce a dashboard that automatically refreshes weekly and features more than 40 separate KPIs across the entire company. These KPIs are continuously displayed around the office, and used by senior leadership and rank-and-file employees alike to guide decisions.[4]

Opportunity Costs

There's no point in having data if you can't turn it into actionable insights.

[4] Read the full case study at fivetran.com/blog/case-study-zoopla

Historically, however, analysts and engineers haven't spent much time analyzing data — instead, they've spent the vast majority of their time building and maintaining sophisticated software to wrangle data. Data science is commonly associated with cutting-edge predictive modeling and machine learning, but about 80% of an average data scientist's time is spent finding and integrating data rather than analyzing it.[5]

CAUTION: *Simpson's paradox (Figure 2.0) is an excellent example of how the same data, transformed in different ways, can lead someone to radically different, and sometimes wholly opposite, conclusions.*

In a nutshell, Simpson's paradox is the phenomenon in which trends and patterns look very different depending on how data is divided or combined.

A similar concept is illustrated by Anscombe's quartet (Figure 2.1): four very different data sets with identical mean, variance, correlation and R2. Simpson's paradox and Anscombe's quartet are powerful reminders that stopping your analysis with basic summary statistics is at best naive and at worst highly deceptive. You must take pains to visualize your data, consider how your data is categorized, and consider lurking variables that might complicate the answer.

The Traditional Approach to Data Integration: ETL

The traditional approach to data integration, known as extract-transform-load (ETL), has been predominant since the 1970s. It is the industry standard among established organizations, and the acronym ETL is often used colloquially to describe data integration activities in general. ETL evolved at a time when computing power, storage and bandwidth were scarce and expensive. The technical shortcomings of ETL, born of that severe resource shortage, look increasingly anachronistic in the era of cloud technology.

[5] infoworld.com/article/3228245/the-80-20-data-science-dilemma.html

Figure 2.0
Simpson's Paradox

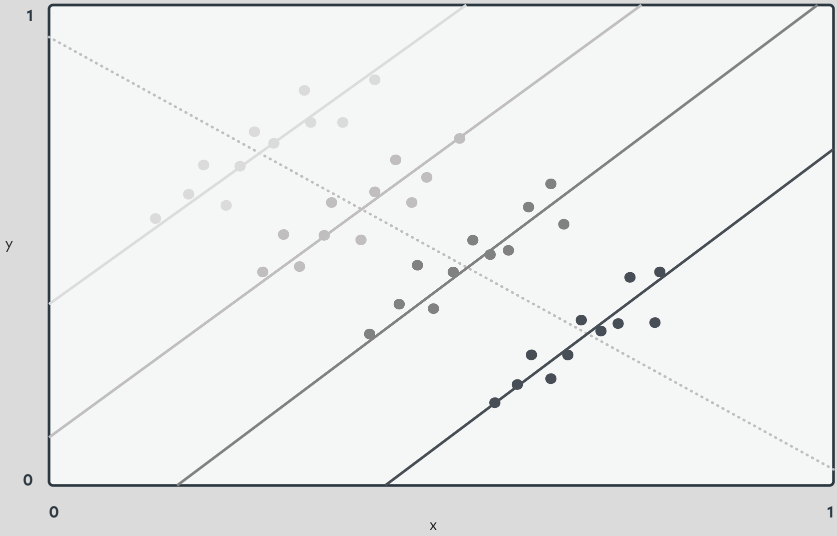
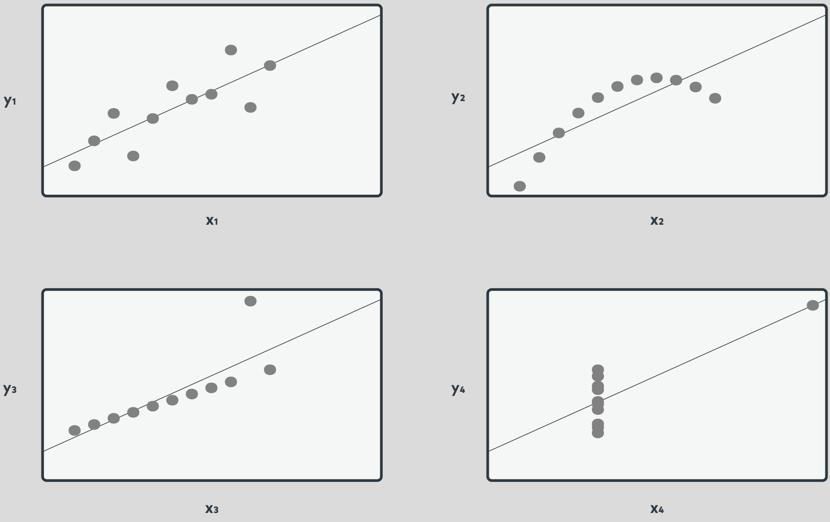


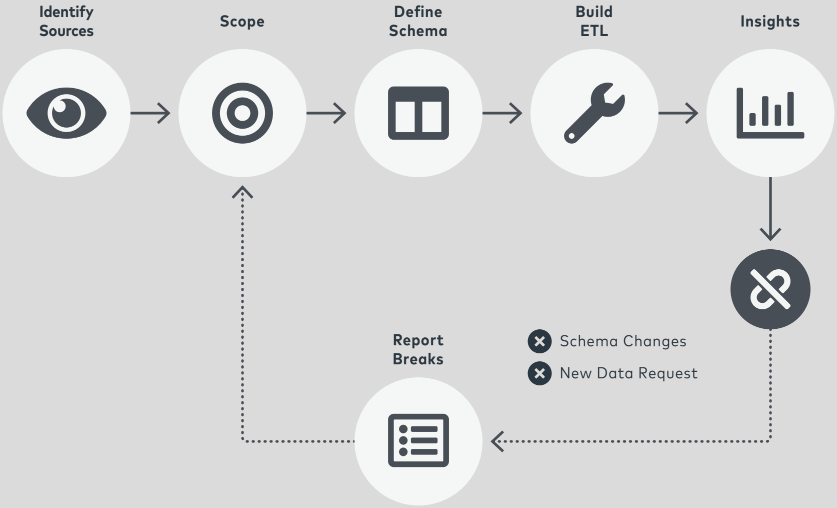
Figure 2.1
Anscombe's Quartet



ETL Workflow

The workflow that engineers and analysts must perform to produce an ETL pipeline looks like so:

Figure 2.2
Data Integration and Analytics Workflow



1. **Identify sources** – apps, event trackers or databases

2. **Scope** – determine the bounds and business goals of the report

3. **Define schemas** – model the data and determine the necessary transformations

4. **Build ETL** – write the software, specifying the details of the API endpoints to call, how to normalize the data, and how to load it into the destination

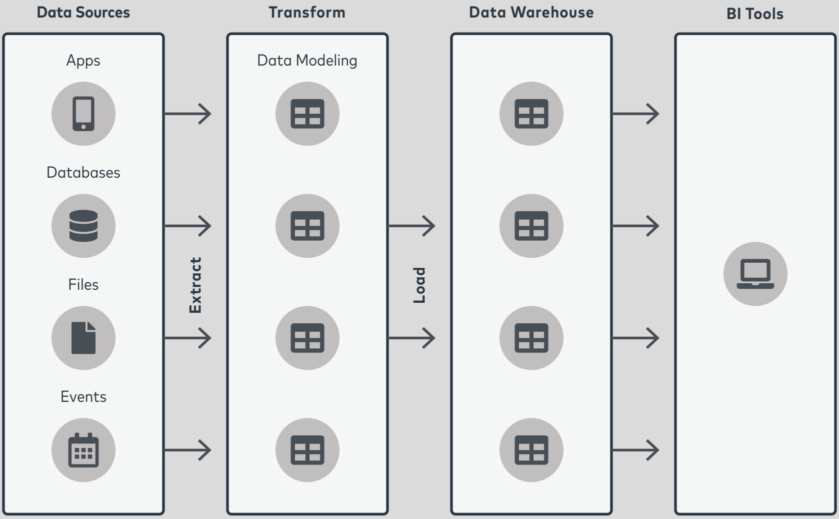
5. **Surface insights** – generate reports that are digestible for key decision-makers

6. **Report breaks** – stoppages leave end users without timely data and cause downtime as a result of:
   a. Schema changes upstream
   b. New data requests made as analytics needs change

7. **Re-scope the project**

The ETL system performs the following steps:

Figure 2.3
ETL



1. **Extract** – data is extracted from connectors
2. **Transform** – through a series of transformations, the data is rearranged into models as needed by analysts and end users
3. **Load** – data is loaded into a data warehouse
4. **Visualize** – the data is summarized and visualized through a business intelligence tool

Orchestration and transformation before loading impose a critical vulnerability on the ETL process. Transformations must be specifically tailored to the unique configurations of both the original and the destination data. This means that upstream changes to data schemas, as well as downstream changes to business requirements and data models, can break the software that performs the transformations.

Since ETL does not directly replicate data from each source to the data warehouse, there is no comprehensive repository of record for analytics. Failures at any stage of the process will render the data inaccessible to analysts and require engineering effort to repair.

Limitations of ETL

Overall, the traditional ETL process has three serious and related downsides:

1. **Complexity**. Data pipelines run on custom code dictated by the specific needs of specific transformations. This means the data engineering team develops highly specialized, sometimes non-transferrable skills for managing its code base.
2. **Brittleness**. For the aforementioned reasons, a combination of brittleness and complexity makes quick adjustments costly or impossible. Parts of the code base can become nonfunctional with little warning, and new business requirements and use cases require extensive revisions of the code.
3. **Inaccessibility**. More importantly, ETL is all but inaccessible to smaller organizations without dedicated data engineers. On-premise ETL imposes further infrastructure costs. Smaller organizations may be forced to sample data or conduct manual, ad hoc reporting.

The Emergence of Cloud Technology

Even a casual observer of technological trends knows that computation, storage and bandwidth have become cheap and ubiquitous. With advances in computing, the cost of computation has plummeted over time (Figure 2.4).

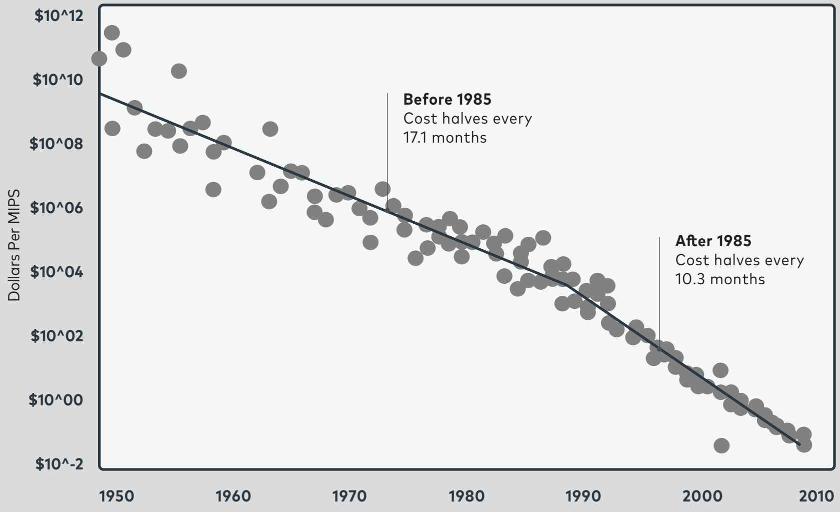
Likewise, in a span of about 35 years, the cost of a gigabyte has plummeted from nearly $1 million to a matter of cents (Figure 2.5).

One effect of these radical cost reductions is that data warehouses can accommodate much larger volumes of data. Organizations no longer need to pre-aggregate and, in the process, discard a great deal of source data. This enables analysts to perform deeper and more comprehensive analysis than ever before.

Although the World Wide Web did not exist until 1991, the cost of internet transit has also decreased radically. In less than twenty years, it dropped from about $1,200/Mpbs to a matter of cents (Figure 2.6).

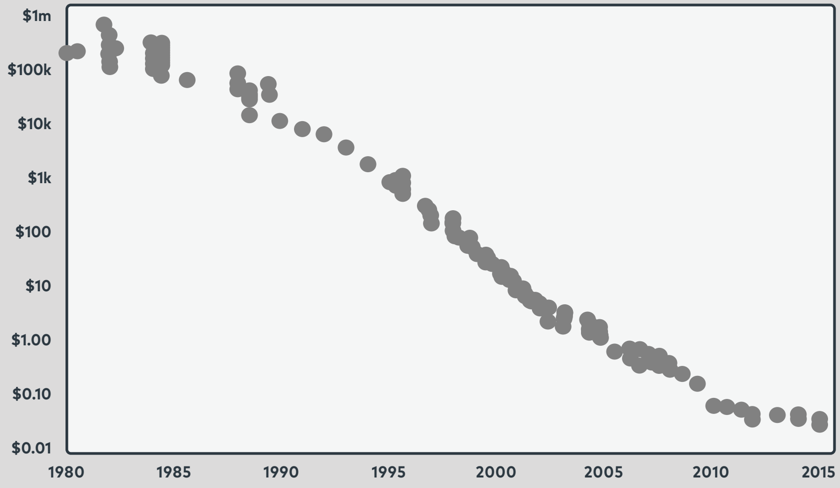
The convergence of these three cost-reduction trends led to the cloud — namely, the use of remote, decentralized, web-enabled computational resources. Cloud technology, in turn, has given rise to a huge range of cloud-native applications and services.

Figure 2.4
Cost of Computing



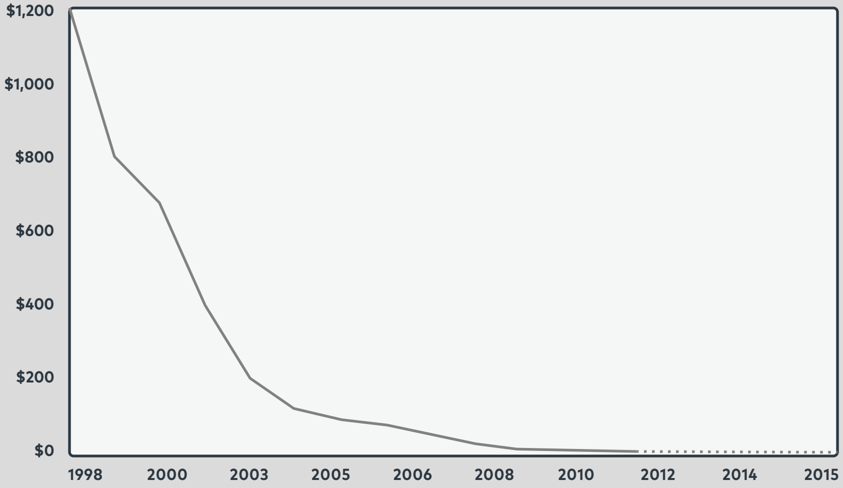
Source: https://frc.ri.cmu.edu/~hpm/book97/ch3/processor.list.txt

Figure 2.5
Hard Drive Cost per Gigabyte



Source: mkomo.com

Figure 2.6
Price of Internet Traffic (Bandwidth)



Source: dr1peering.net

TIP: *Cloud-native apps and services encompass every kind of business activity, including customer relationship management, billing and payments, ecommerce, email marketing, benefits administration, project management and customer service. There is a very good chance your organization already uses several such services.*

A major benefit of the cloud is that analysts and end users of data are no longer shackled to physical infrastructure. Instead, they can host services on the web. Issues of scale and accessibility become much easier to resolve. Organizations can add or remove computation and storage resources on the fly, and users can access dashboards and reports through any web-enabled device.

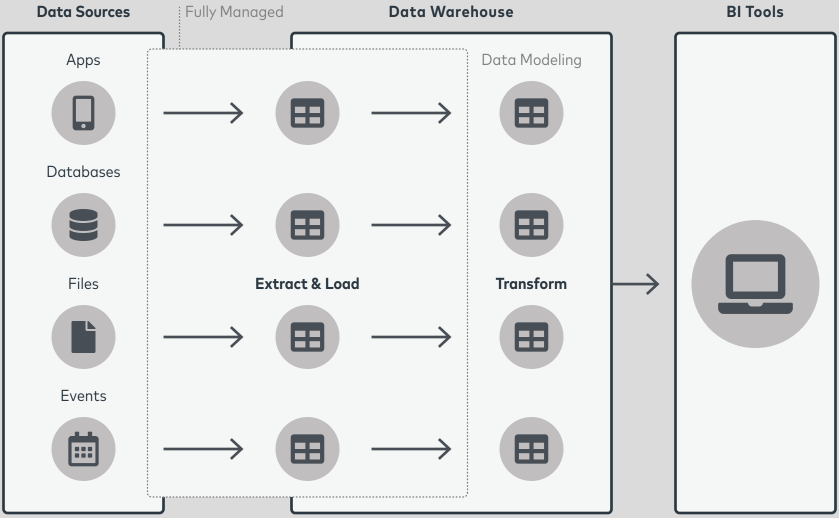
CAUTION: *Let's not overstate the capabilities of the modern web. We do not (yet) have the technology to instantly upload and download terabytes of data around the world. At the largest scales, it might take less time to physically ship a container full of hard drives than to move the data over the internet. It is still important to compress data and minimize the volume piped from source to destination.*

The Modern Approach to Data Integration: ELT

The same trends that enabled the growth of the cloud — the plummeting costs of computation, storage and bandwidth — also enable organizations to circumvent the problems posed by ETL. Specifically, data can be streamed and loaded before it is transformed. This revised sequence is extract-load-transform (ELT) and is the modern successor to ETL (Figure 2.7).

What we refer to as the "modern data stack" is predicated on ELT and replaces on-premise technologies with cloud-native SaaS technologies. Such a setup facilitates automation, collaboration and scalability, and avoids many of the costs of an on-premise stack. Properly implemented, the modern data stack delivers continuous data integration and organization-wide accessibility, with a minimum of manual intervention and bespoke code.

Figure 2.7
ELT



Switching the order of the loading and transformation stages addresses each of the three major shortcomings of ETL:

1. **Complexity**. The pipeline is simplified — by delivering standard schemas to a warehouse first, without custom transformations, a great deal of pipeline-

related work is shifted downstream to analysts instead of data engineers.

2. **Brittleness**. The pipeline is more resilient and less risky — because transformations are applied after the data is warehoused, breakages caused by changes in source systems mainly affect the analytics layer. Analysts can typically remedy these issues without the help of data engineers.

3. **Accessibility**. The pipeline is more accessible because it's less labor-intensive to maintain. Because the pipeline is greatly simplified and intrinsically more resilient, third parties can build and maintain a standardized tool for multiple customers, as well as derivative products to enhance analytics efforts. Purchasing a standardized tool essentially outsources and automates the extract and load stages.

In-warehouse transformations enable the creation of derivative tables, called "views," without altering the source data. This allows organizations to create a repository of record that is immune to changing business needs or upstream schema changes. The same data can be applied to multiple use cases.

ELT also reduces the workload of engineers. Once the data is warehoused, analysts can use SQL to perform transformations at their discretion. Analysts might still require complex transformations that must be carefully orchestrated and planned, but stoppages and failures will no longer cripple the entire data pipeline or consume significant engineering resources.

A Better Way Forward: Automated ELT

The simplified, cloud-based nature of an ELT data stack lends itself easily to automation and outsourcing.

The specific activities involved in automated ELT include detecting and replicating data changes, lightly cleaning and normalizing data, and updating and creating tables. These activities require a deep knowledge of data sources, extensive data modeling and analytics expertise, and the engineering know-how to build robust software systems. Without an automated data integration tool, your team must perform these activities and develop the requisite capabilities.

By contrast, automating and outsourcing ELT allows you to leverage the expertise of outside parties who understand every idiosyncrasy of the underlying data sources — and have stress-tested their connectors against a much wider range of corner cases than you likely ever will. The main benefits of automated ELT, as with most forms of automation, are savings of time, effort and money. Your data or business intelligence team should focus on providing actionable insights, not on routine, upstream work focused on problems that have already been identified and solved.

Data engineers can leverage the time savings of automated ELT to shift their efforts toward problems impacting external customers, or to pursue higher-value data activities such as machine learning and artificial intelligence. Automated ELT is best thought of as a force multiplier rather than as a replacement for human talent.

The radical accessibility that automated ELT offers is exemplified by the typical automated data integration workflow:

1. **Sign up** – Activate your account.
2. **Select** – Choose your sources and data warehouse.
3. **Authenticate** – Activate your connections using your existing credentials.
4. **Automate** – Let the system take care of historical sync and ongoing changes.

Although a full historical sync may take hours or days depending on the volume of data hosted by the data source, the steps that actually require human intervention may take no more than a few minutes.

Another benefit of an automated data integration tool is that every company using a particular data source must solve exactly the same problem, so the tool provider can offer every customer a single, standardized solution with the exact same schemas. These standardized schemas enable the creation of derivative products that bolster analytics efforts. Users of the same integration tool will have access to the same SQL-based transformations, embedded analytics products and BI tool modules. ELT effectively brings the benefits and scale of modular, interchangeable parts to analytics.

A final benefit of automated ELT is that you can outsource cybersecurity and regulatory compliance. The policies, procedures and technologies that prevent malicious or illegal access to your data require deep expertise to develop, and you are far better off trusting a credible outside party than attempting to build your own solution.

Empowering engineers, analysts and end users through automation and self-service does, however, increase the importance of data governance. Access and transparency can be extremely valuable, but must be managed through stringent auditing, documentation and assigning of permissions. As an organization matures, the analysts who typically build dashboards and reports from data might find their roles shifting toward data governance, while end users increasingly generate reports and dashboards themselves.

CASE STUDY: DocuSign Uses Automated Data Integration to Triple Number of Data Sources

DocuSign is the world leader in e-signature technology, helping individuals and organizations automatically prepare, sign, act on and manage agreements.

Formerly, DocuSign used SQL Server as a data warehouse, with a set of six data sources managed by an engineer. These homespun connectors took three to six months to build and up to 20 hours a week of engineering time to maintain. This workload became untenable as the company continued to grow, especially as engineers were needed for core projects, and business teams needed to model and catalogue data from applications.

With the help of an automated data integration solution and a more elastic cloud data warehouse, DocuSign was able to save all 20 hours of engineering time and triple the number of its data sources from six to 18. The sudden increase in scale and savings of time and labor have accompanied another highly positive development — people from all teams across the company now use over 100 active dashboards in their BI tool.[6]

REMEMBER: *Cloud technology has created both a profusion of valuable data and the tools required to adequately handle it. ELT overcomes many of the disadvantages of ETL, making data and analytics far more accessible and scalable than ever before.*

[6] Read the full case study at fivetran.com/blog/case-study-docusign

Chapter 3: Why You Shouldn't Build Your Own Data Pipeline

IN THIS CHAPTER:

- How to estimate the monetary and non-monetary costs of building your own data pipeline
- Convincing your organization to adopt an off-the-shelf solution

Key Considerations

If the modern data stack radically simplifies data integration, is it worthwhile for your organization to build its own ELT pipeline, even in the cloud?

> **CAUTION:** *Recall that manual data integration is not scalable and that ETL has been made obsolete by prevailing technological trends, as discussed in Chapter 2. Here, we are primarily discussing the construction of a bespoke ELT pipeline, although the following arguments will apply if an organization attempts to build a custom ETL workflow as well.*

Time and Monetary Cost

As we mentioned in Chapter 2, roughly 80% of an average data scientist's time is spent constructing data pipelines — a task for which most data scientists have limited aptitude, interest or training (Figures 3.0 and 3.1). The most obvious argument against constructing your own ELT pipeline is the cost of building and maintaining it, in terms of time, money, morale and lost opportunities.

Suppose your organization needed a complement of five connectors for customer relationship management, customer support ticketing, advertising automation, project management and subscription billing.

Each of the five connectors takes about five weeks for an engineer to build, or five person-weeks (pw):

(5 connectors) * (5 pw)

Each connector will likely need a dedicated week of maintenance work per quarter, adding up to four weeks per year:

(5 connectors) * (5 pw + 4 pw)
(5 connectors) * (9 pw) = 45 pw

That makes 45 weeks out of 52 weeks in a year. Assuming a slightly generous vacation or sick day policy, that is essentially a year's worth of work for a software engineer who costs about $120,000 before accounting for benefits (Figure 3.2).

In subsequent years, your engineer will continue to update each quarter (four weeks) and handle bugs and edge cases as they crop up (one week), for a total of five pw per connector.

(5 connectors) * (5 pw) = 25 pw

That makes 25 weeks out of 52 weeks in a year, all dedicated to ongoing maintenance. Let's ballpark the cost to half the engineer's yearly salary, or $60,000.

Purchasing or outsourcing five connectors will likely cost far less than either of the figures above. These costs will, of course, scale in direct proportion to the number of data sources you use.

Figure 3.0
What Do Data Scientists Spend the Most Time Doing?

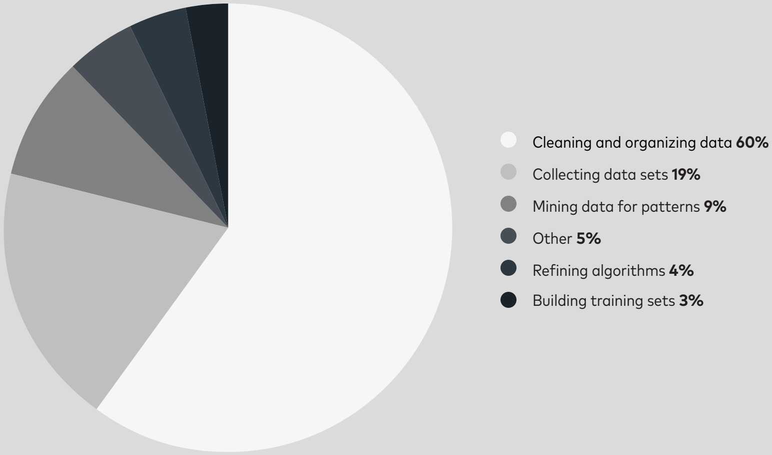
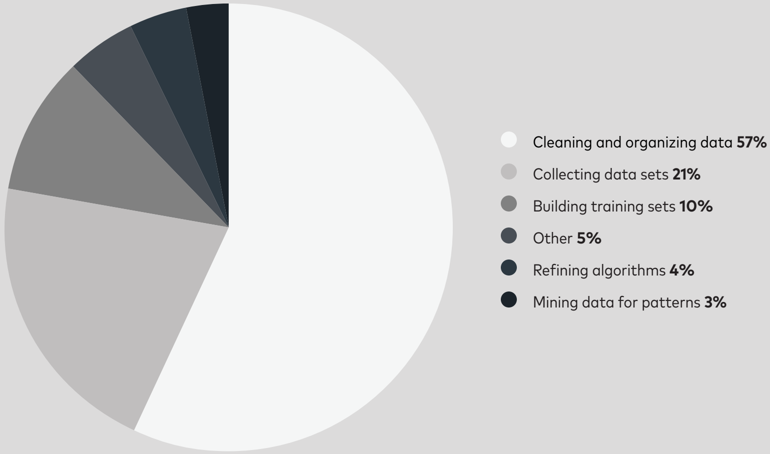
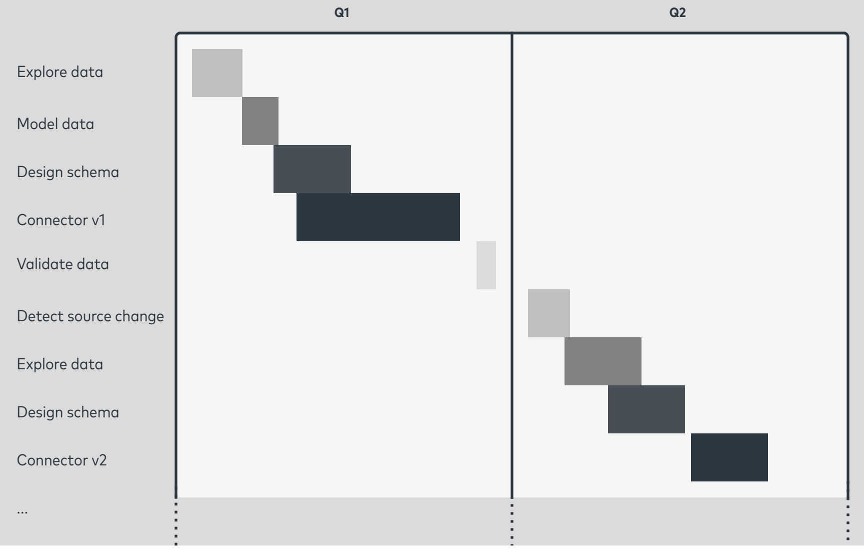


Figure 3.1
What Is the Least Enjoyable Part of Data Science?



Source: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/

Figure 3.2
How Much Work Goes Into Building Your Own Connector?



The sample Gantt chart above demonstrates the cyclical, recurring nature of data engineering work, even under an ELT framework, as upstream schemas continue to change.

Morale

If you want to keep your analysts, engineers and managers happy, consider the following problems associated with building your own connectors or manual reporting:

1. Diversion from other software engineering, data science or analytics duties — a very common irritant among new data scientists at understaffed organizations, which can lead to turnover
2. Frustration and exhaustion from the complexity of maintaining data integrity, particularly by persons lacking the appropriate training
3. Downtime caused by continually increasing complexity as additional sources of data are inevitably added
4. Misguided decisions caused by lags between requests for business intelligence and delivery of actionable insights — insights that might be stale by the time they arrive

For most data professionals, database maintenance is a chore, not an aspiration.

Learning Curves

The five-week estimate cited above applies to APIs that are relatively straightforward. Not all APIs are so tractable. Some ignore best practices, some are poorly documented, and some are just very complex.

Data from an enterprise resource planning (ERP) tool, for instance, might encompass every imaginable business activity, represented in dozens or hundreds of individual tables with complex interrelations. It can take many iterations to build a mature piece of software around such a data source, multiplying the costs above.

Complexity at Scale

It is highly unlikely that your organization's data needs will stop at five connectors. As mentioned earlier, a typical company now uses more than 100 apps,[7] and this figure is likely to climb. It's hard to justify expanding your team's obligations when you can cost-effectively outsource pipeline engineering.

Standardization

The final argument against constructing your own data pipeline is that connectors built by an outside party become robust through testing against dozens of corner cases from a wide range of customers. These connectors produce standardized data sets with standardized schemas (Figure 3.3).

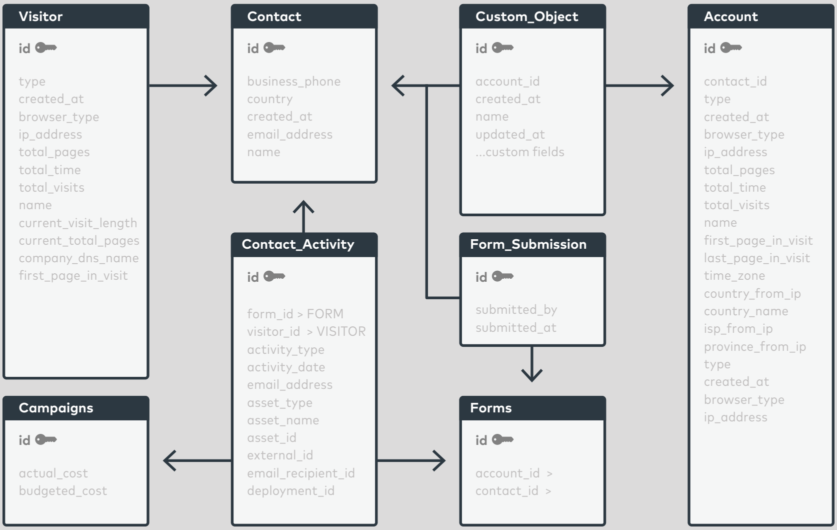
In practice, you are unlikely to use normalized tables to directly power your dashboards, and will instead transform them into models that are more appropriate for their end users.

In principle, this also allows any organization that uses the same connectors to leverage the same transformations, because the data is all structured exactly the same way. Plug-and-play recipes or templates of this kind can be written in SQL or languages specific to a BI platform, such as LookML.

In general, standardization creates economies of scale, as the provider comes to understand every idiosyncrasy of the underlying data source and shares the benefits of that understanding with its customers.

[7] https://www.wsj.com/articles/employees-are-accessing-more-and-more-business-apps-study-finds-11549580017

Figure 3.3
Standardized Schema



TIP: *There are as many ways to build a schema from a data set as there are opinions and use cases. One approach to maintaining a replicable, easily understood standard is to normalize. A full discussion of the normal forms is beyond the scope of this guide, but the key idea is that normalization creates a relational structure that eliminates redundancy and inconsistent relations between data elements.*

REMEMBER: *The concept of extracting and loading data seems simple, but data pipelines are highly sophisticated pieces of technology that require a great deal of skill, labor and attention to detail to successfully build. You are almost certainly better off shopping around for a solution than attempting to build and maintain one in-house.*

Making the Case for Buying

You might encounter some degree of resistance in your organization if you propose buying a data pipeline tool. Data engineers might find the prospect of automating some of their work threatening, and executives who are several degrees removed from the challenge might not immediately see the benefits.

CASE STUDY: Papier Uses ELT to Double Its Data Sources and Build a Customer Attribution Model

Papier is a design and personalization company selling stationery, invitations, cards and photo books. Founded in 2015, the company has flourished in its native UK and is currently expanding both its operations and product portfolio. The complexity of its data integration efforts have grown along with its business.

Papier originally centralized its ad, clickstream and transactional data using ETL scripts and tools developed in-house. This quickly became unscalable in terms of sheer workload and corrective efforts for inaccuracies and inconsistencies often required re-syncs.

After engaging the services of an ELT data pipeline, Papier was able to automate its data ingestion workflow and double the number of data sources it used in its analytics. Syncs that used to be performed once a day now occur every hour. Most importantly, by centralizing its data and doubling its number of data sources, Papier was able to build a more robust customer attribution model, determining which ads and other marketing activities were most responsible for its revenue.[8]

Winning Over Engineers

Engineers sometimes favor customization, fine-grained configurability and control over accessibility. These talking points might help you win them to your cause:

How Building Pipelines Hurts You

1. Complexity and long turnaround times produce painful bottlenecks.
2. Without the ability to quickly make informed decisions, companies fail to adapt and fall behind the competition.
3. Data busywork and chores are no fun.

How Outsourcing Helps You

1. You'll no longer be a bottleneck.
2. Outside vendors who specialize in data connectors are far more knowledgeable and experienced at solving these problems. They're likely to have considered far more corner cases than you and will do a better job.

[8] Read the full case study at fivetran.com/blog/case-study-papier

3. ELT is a force multiplier. You can now manage a much larger range of connectors with a lot less effort, making it easier to meet your organization's expectations.
4. Fully managed ELT is incredibly intuitive and requires almost no training or instruction. You'll have the time to do things other than learn highly tool-specific skills.
5. You won't have to keep hiring and managing more people for data engineering.
6. You'll have the opportunity to pursue highly strategic activities, e.g.:
    a. Custom tools and infrastructure for analysts
    b. Infrastructure to support AI and machine learning
    c. New software products

The best way to approach this challenge is to focus on what engineers would rather spend more of their time doing. In practice, very few engineers aspire to spend most of their time writing data connectors. They would rather pursue higher-value projects than build and maintain infrastructure.

Convincing Your Boss

Executives who are a few degrees of separation away from data engineering may need to be convinced that the ultimate value of a new tool justifies both its premium price and the personnel reshuffling that will likely ensue.

Highlight Other Companies' Successes

Luckily, other companies' experiences offer plenty of lessons. You may want to work backward in stages, discussing the benefits of improved business intelligence and actionable insights first, and the prerequisites of realizing such improvements later.

Discuss Common Pain Points and Key Benefits

Common pain points:

1. Manual reporting has a very long turnaround time and is only marginally better than flying blind.
2. Different business units have separate data silos and struggle to share relevant information.
3. In-house tools become unsustainable as new data sources are added, data volume is increased, and performance requirements become more stringent.
4. Legacy databases and on-premise data warehouses are reaching their

performance and usability limits.

5. Engineers have better things to do than maintain databases.

Common benefits:

1. Time savings – dramatically shorter turnaround between reports
2. Data gains – massive expansion of data availability and timeliness
3. Quality gains – data is more comprehensive and timely
4. Culture gains – data access and data-driven decisions democratized across the company
5. Labor savings – less engineering time on bespoke integration tools and database maintenance; analysts don't have to manually assemble reports
6. New insights and products – relocating engineering time away from data integration means more bandwidth for exploring opportunities and developing products

Remember that executives are generally downstream of the data integration process (unless they're CTOs), so it's important to emphasize the benefits of good BI before discussing the technicalities of data warehousing and data integration.

CASE STUDY: MVF More Than Doubles Its Monthly Rescued Revenue Using a Modern Data Stack

MVF is a customer-generation platform that delivers sales leads on a pay-per-lead basis. Without a modern data stack, MVF did not have a single source of truth, with separate databases and ad hoc data stacks spread across teams. It took two or three weeks to generate reports.

With Fivetran, MVF was able to centralize its data and readily identify unsold leads. MVF doubled its monthly revenue from unsold leads from £300k a month to around £700k. The reports that used to take two or three weeks are now automatically and continuously generated.

In addition, their engineers have been entirely spared the initial effort of spending four to eight weeks building a connector and the ongoing costs of maintenance and debugging. The automation of data integration has also allowed MVF to add eight more data sources. In turn, the data engineers now pursue higher-value, more strategic projects.[9]

[9] Read the full case study at fivetran.com/blog/case-study-mvf

REMEMBER: *Your organization's long-term health depends on keeping up with the technological curve. Data literacy has become critical to market competitiveness, and obstruction by executives or recalcitrance by engineers may ultimately imperil the entire organization.*

Chapter 4: Business Considerations for Choosing a Data Integration Tool

IN THIS CHAPTER:

- How does pricing work?
- Does a tool fit your organization's needs?
- Future-proofing

Whether or not an automated data integration solution will save time, money and labor depends on your organization's size and maturity, as well as the particular characteristics of the data pipeline provider.

At very small scales, your organization might not need a data pipeline, especially if you are an early-stage startup that only uses one or two data sources, or if you are only conducting qualitative research while trying to find product-market fit. Conversely, your organization might have a niche use case with extremely stringent performance, security or regulatory requirements. Certain data science applications can be extremely sensitive to even nanoseconds of latency.

Excluding the above scenarios, your organization probably struggles with the high engineering costs of building and maintaining data connectors, or endures long report turnaround times from connector maintenance and manual reporting. If so, you should explore the business case for buying a data integration solution.

How Do Pricing and Costs Work?

Familiarize yourself with the pricing structures of prospective tools. Here are some common pricing models:

- **A flat subscription fee**, which might have higher fixed costs in exchange for cost predictability.
- **Pricing by volume** of data, as counted in gigabytes or rows. A volume-based pricing model can be highly advantageous if you currently handle a very modest scale of data but want to test out a new tool over an extended period, or if you plan to gradually move your workflow to the new system.
- **Per-seat pricing vs. a single fee** for your organization. Per-seat pricing models will typically cost less at small head counts, but are more of an administrative hassle. Single fees for an entire organization can be simpler and cost less at larger scales.

You might also encounter combinations of pricing models. A service might have a flat "platform" fee and then additional fees for each unique data connector. Volume-based rates might vary by connector. Providers might offer a freemium model up to a certain data volume, or with a restricted feature set. In other words, your mileage may vary.

Does a Tool Fit Your Team's Skills and Future Plans?

Other factors to consider include the trade-off between ease of use and configurability, and compatibility with your team's existing skill set.

Non-technical users might or might not be familiar with SQL, but they can almost certainly navigate a BI tool. Analysts typically know SQL, statistics and perhaps a scripting language like Python. Data scientists might have a deeper technical skill set that includes more advanced statistical training and additional languages like Java, as well as "big data" technologies such as Hadoop or Spark. Engineers will likely be familiar with a range of high- and low-level computer languages, as well as an assortment of technology platforms.

Different data integration tools feature different levels of complexity and accessibility. Some rely heavily on custom scripting and offer only the basic scaffolding on which you build your own data pipeline. Others offer drag-and-drop GUIs that allow relatively non-technical users to orchestrate data replication and transformation, but these have two clear drawbacks: a steep, highly platform-specific learning curve and the automatic generation of spaghetti code. Still others combine completely automated data replication with version-controlled, SQL-based transformations.

The trade-off will boil down to accessibility versus configurability. If your goal is to promote data literacy across your organization, then you should find a tool with the lowest barrier to use and the broadest applicability to different use cases. For more specialized use cases, less accessible but more powerful and configurable tools optimized for specific niches might be appropriate.

Vendor Lock-In and Changing Needs

Before you commit to a contract, consider whether the tool will serve your needs in the future. Ask the following questions:

- Does the tool feature the connectors you currently use or anticipate using?
- Can you easily enable additional features or add data connectors to your account as needed?
- Are connectors consistently updated to keep abreast of upstream API changes, and are these updates accompanied by changelogs specifying changes?
- Are new connectors regularly added to the tool?
- Is the support team responsive and capable of keeping up with product changes and your changing needs?
- Can you export data models and transformations from one platform to another, or will you have to reverse-engineer and rebuild them if you ever switch to a new tool?

Future-proofing is important because switching platforms can be very costly and disruptive.

TIP: *Although SQL comes in a number of dialects, it is an industry standard in analytics. Data models and transformations written in SQL should, in principle, be easy to port from one system to another. By contrast, procedures, data models and transformations stored in proprietary file systems or languages cannot be easily ported, and introduce a serious risk of vendor lock-in.*

REMEMBER: *To evaluate the business suitability of a certain tool, look at the total cost of ownership and whether the tool spares you potential organizational complications, such as changing needs, resilience to accidents and failures, and regulatory compliance.*

Chapter 5: Technical Considerations for Choosing a Data Integration Tool

IN THIS CHAPTER:

- ETL vs. ELT
- Evaluating data connector quality
- Understanding how automation works in your stack

Once you have identified a business need for a data integration solution, you should consider the technical characteristics of each tool.

Data Connector Quality

The basic component of every ELT data pipeline is the data connector. A data connector ingests data from an API or database log, performs some light cleaning and normalization, and then loads it into a data warehouse. When evaluating data connector quality, consider the following:

- **Open-source vs. proprietary**. As with other pieces of software, there is a trade-off between voluntary, crowd-sourced talent and dedicated professional attention. Overall, there are more open-source connectors for a wider range of data sources, but proprietary connectors tend to be of higher quality and integrate more seamlessly with other elements of a data stack. In particular, providers of proprietary technology have incentives to apply strict QA, maintenance and engineering principles.

- **Standardized schemas and normalization**. Data from API feeds is not usually provided in normalized form. Normalization fosters data integrity by eliminating redundancy and establishing clear, consistent relations between tables. For a given data set, there are as many opinions as there are possible schemas, but only a handful of possible normalized schemas. Since there are only a few ways to normalize a data set, normalization also lends itself to schema standardization, which introduces economies of scale that benefit all users.

TIP: *Make sure to inspect providers' documentation for entity relationship diagrams (ERDs), which illustrate schemas. ERDs should clearly communicate the fields that are available in every table and the relationships between tables. Your analysts should be able to tell if the schema contains useful fields and is normalized.*

- **Incremental vs. full updates**. What is the connector's replication strategy? The initial sync will require querying either the full data set or a large subsection of it, but subsequent updates should not. Does the connector update incrementally using logs or other forms of change detection, or does it query full data sets every time it syncs? Incremental updates allow more frequent, lower-volume updates. Frequent full replications of operational databases introduce the additional danger of interfering with critical business operations.

Support for Sources and Destinations

Different data pipeline tools will support different data sources and data warehouses. Make sure the tool you are evaluating supports those that are important to you. If not, does the provider offer a way for customers to suggest new sources and destinations? Do they routinely add new ones?

For that matter, can the tool support multiple sources and destinations? Your organization may eventually use dozens or hundreds of connectors for the same type of data sources if, say, you handle many advertising accounts on behalf of your clients, or your company goes through a merger or acquisition and must combine data from multiple accounts and platforms. You may also opt to sync with multiple data warehouses in the interest of redundancy.

Finally, consider whether and to what degree the tool accommodates custom data integrations. You may need to integrate data from obscure data sources that are unsupported by a standard off-the-shelf connector. Does the tool you are evaluating support cloud-based functions that allow you to combine custom connectors written by your engineers with the rest of your infrastructure? In a pinch, does the tool support ad hoc loading and warehousing of data from CSV or JSON?

Configuration vs. Zero-Touch

Highly customizable and configurable tools allow users to tweak every last parameter and design the exact workflow they desire. This approach requires engineers who are proficient in scripting languages, highly experienced with orchestration, and good at building robust software. It also requires engineers to deeply understand each individual data source, or to collaborate closely with analysts in order to explore, understand and model the data. Schemas must eventually become usable data models, and designing a good schema, as well as a good progression from raw to refined data, is difficult and can be as much art as science.

Under a highly configurable approach, users must correctly configure and maintain the data integration software. This includes reconfiguring pipelines whenever downstream business needs and upstream data sources change. The highly configurable approach is best suited to organizations with deep technical talent pools that actively want to pursue these challenges, and are supremely confident in their ability to deliver better, more reliable results than an off-the-shelf product.

⚠ **CAUTION:** *There are also GUI-based data integration tools that allow non-engineers to visually program orchestrations and transformations. Instead of a highly technical team of engineers, you will need analysts or end users who are highly familiar with a proprietary visual programming language. This may introduce serious problems with overspecialized skills and vendor lock-in.*

By contrast, zero-touch, fully managed tools are extremely accessible due to their set-and-forget nature. From the customer's perspective, the connectors are standardized, stress-tested and maintenance-free. Maintenance and future iterations of the connectors become the service obligation of experts who understand every idiosyncrasy of the underlying data and have tested their connectors against a wide range of corner cases.

In place of orchestrating and transforming before loading, transformations can be scheduled and performed by analysts using SQL. This makes the zero-touch approach far more appropriate for companies that don't have access to a deep pool of top-shelf engineering talent to handle building and maintaining pipelines, and want to use their engineering talent for other high-value projects.

TIP: *Self-service is another important consideration. Consider whether and to what degree the tool you are evaluating offers the ability to start an account without the intervention of an account manager or customer support person. Self-service may put slightly more of an onus on your team but can also allow you to start and stop a subscription more easily.*

Automation

The purpose of modern data integration tools is to remove as much manual intervention and effort from the process as possible. To that end, consider the following labor-saving tools and features of automation:

- **API**. It can be extremely helpful to programmatically control the tool so that administrative functions and other chores can be performed automatically rather than by hand. Such functionality can be especially helpful when large numbers of people need varying levels of control over the tool or if you build products on top of data integration.
- **Handling data type changes**. Upstream schema changes can alter the type of a particular value, i.e., from integer to float. An automated tool must be able to reconcile old and new data types without human intervention.
- **Continuous sync scheduling**. Data from these connectors should either stream into your data warehouse or sync at short, regular intervals. Determine how often your organization needs data to be updated and set it and forget it.
- **Automatic schema migrations**. Schemas will inevitably change as additional data elements are added to a data set. Does the connector automatically accommodate these changes with a minimum of disruption to anything downstream, i.e., without deleting any tables or fields? Does the connector avoid full re-syncs whenever possible?
- **General performance**. Finally, you should consider a number of characteristics that will determine possible downtime for your system, including:
    - How long does an initial sync take?
    - Is the data updated incrementally, or is a full sync required every time?
    - Which conditions trigger a full sync?
    - How often is the data updated, and how does it match your needs?
        - Does it stream live? Every few minutes? Once a day?

The answers to these questions can influence costs imposed by downtime and infrastructure usage — costs that might not be included in a pipeline tool's formal pricing structure, but can be a significant burden on your organization.

Transforming Within vs. Before the Data Warehouse

Under ELT, transformations are performed in an elastic, cloud-based data warehouse. Elasticity — and the separation of compute and storage — allows resources to be scaled up and down as needed. This eliminates the need to forecast hardware requirements and buy excess capacity.

By contrast, ETL, whether cloud-based or on-premise, requires a data architecture with an additional stage in the data stack to handle transformations before they are loaded. If the data stack is on-premise, the data warehouse itself can constrain the volume of data that is loaded, making transformations necessary in order to limit the volume and flow of data.

The fundamental advantage of ELT and transformations performed within the data warehouse is that they are non-destructive — that is, they allow the underlying data to remain completely undisturbed while additional tables featuring the desired models are created. This means that failed transformations have no permanent consequences and can be repeatedly attempted. Likewise, analysts can adjust models to changing business requirements without losing any data.

A final benefit of performing transformations within the data warehouse is that the transformations can be written in SQL, making them accessible to analysts. Analysts commonly create views within a data warehouse in order to consolidate or modify tables. ELT tools that support in-warehouse transformations give analysts the ability to create views in a systematized manner.

Recovery From Failure

Bugs and errors will inevitably crop up over the course of data integration, and data integrations will inevitably fail. The last thing you want is to accidentally and permanently lose data as a result.

An important characteristic of a data integration tool is *idempotence* — the ability to repeatedly attempt the same process and produce the same result each time. This is especially valuable with complicated, multi-step processes, in which the exact point of failure may not be obvious.

Another important principle is *net-additive integration*. When a value is deleted in the source data or a table is dropped, is it retained (but flagged) in the data warehouse? Retaining but flagging a defunct value preserves historical records and is beneficial for audits, recovery from failure, and analyzing longer-term trends and attrition.

TIP: *Make sure to read the service-level agreements (SLAs) of any providers you are considering, and hold them accountable! In particular, make sure the vendor can meet the same SLAs that you would demand from your team. An SLA offers clear expectations around uptime, downtime, speed and volume of data transfer, and other performance metrics.*

Security and Regulatory Compliance

Cybersecurity and privacy are highly fraught subjects, both legally and in the public imagination.

The following is a list of considerations:

- **Regulatory compliance**. Your data integration provider should at a minimum be aware of such standards as GDPR, SOC 2, HIPPA and other relevant regulations. A good tool will support the ability to omit or encrypt personally identifiable information (PII).
- **Owning your data**. Your data integration provider should not access or retain the data for longer than is necessary to replicate it.
- **Roles with varying levels of access**. Not everyone who uses the tool should have unlimited authority to create, delete or modify warehouses, connectors or transformations, or perform other sensitive actions. The tool should feature a range of roles from administrator to read-only.
- **Column blocking and hashing**. As a matter of security and regulatory compliance, you should be able to obscure or omit PII from every table that you sync.

REMEMBER: *Choosing a data integration tool fundamentally boils down to how it can make the jobs of analysts and engineers easier.*

You should consider:

- *Quality of data connectors*
- *Whether the tool supports data sources and destinations you currently use or plan to use*
- *How much hands-on configuration it needs*
- *How well it runs without manual intervention or supervision*
- *When the tool performs transformations*
- *How resilient the tool is to failure*
- *Security and regulatory compliance*

Try out multiple tools! In the next section, we will discuss how to get started.

Chapter 6: Seven Steps to Getting Started

IN THIS CHAPTER:

- Understand your needs and goals
- Make sure you have an idea of what success looks like for you
- Try before you buy, and try hard!

For all that a cloud-first, fully managed data stack promises, it is not appropriate for every organization.

To choose the right course for your organization, you must:

1. Make a thorough assessment of your needs
2. Decide whether to migrate or start fresh
3. Evaluate cloud data warehouse and business intelligence tools
4. Evaluate data integration tools
5. Calculate the total cost of ownership
6. Establish success criteria
7. Set up a proof of concept

Assess Your Needs

There are a few reasons that you might not want to outsource your data operations to a third party or a cloud.

The first and most obvious is that your organization may be very small or operate with a very small scale or complexity of data. You might not have data operations at all if you are a four-person startup still attempting to find product-market fit. The same might be true if you only use one or two applications, are unlikely to adopt new applications, and your integrated analytics tools for each application are already sufficient.

A second reason not to purchase a modern data stack is that it may not meet certain performance or regulatory compliance standards. If you are a high-frequency trading company and nanoseconds can make or break your operations, you might want to avoid third-party cloud infrastructure and build your own hardware.

Otherwise, if your organization is of a sufficient size or maturity to take advantage of analytics, and data refresh cycles of a few minutes or hours are acceptable, proceed.

Migrate or Start Fresh

Data integration providers should be able to migrate data from old infrastructure to your new data stack, but the task is a notorious hassle because of the intrinsic complexity and variety of data. Whether your company decides to migrate or simply start a new instance from scratch depends heavily on whether historical data is deemed sufficiently important.

If your organization has already purchased or contracted for products or services, it may be costly to end those contracts. Beyond money, familiarity with and preference for certain tools and technologies can be an important consideration.

Take care that prospective solutions are compatible with any products and services you intend to keep.

TIP: *A phased approach is perfectly normal. Many companies will spin up a new data warehouse while retaining their old, soon-to-be-redundant data warehouse until all the data and processes are migrated to the new environment.*

Evaluate Cloud Data Warehouse and Business Intelligence Tools

You will have to compare and contrast solutions for every part of the data stack. Before you get a data integration tool, start a little downstream and think about what features you will need in a cloud data warehouse and business intelligence tool.

Cloud data warehouse features to consider include:

1. Centralized vs. decentralized data storage
2. Elasticity – can the data warehouse scale resource use up and down quickly? Are compute and storage resources independent or tightly coupled?
3. Concurrency – can the data warehouse accommodate multiple simultaneous tasks?
4. Load and query performance
5. Data governance and metadata management
6. SQL dialect
7. Backup and recovery support
8. Resilience and availability
9. Security

Business intelligence tool features to consider include:

1. Seamless integration with cloud data warehouses
2. Ease of use and drag-and-drop interfaces – especially helpful if you want to create a data-driven culture across your company
3. Automated reporting and notifications
4. Ability to conduct ad hoc calculations and reports by ingesting and exporting data files
5. Speed, performance and responsiveness
6. Modeling layer with version control and development mode
7. Extensive library of visualizations

Make sure any data warehouses and BI tools you evaluate are compatible with each other. It also pays to carefully review a range of perspectives on different tools.

Publications like Gartner often aggregate such information. Make sure you read before you leap!

Evaluate Data Integration Tools

As we covered earlier, there are many important characteristics to consider with regard to data integration tools.

A short list of what you should look for:

1. Customization and configurability vs. ease of use and accessibility
2. Reliability and performance of the software
3. Quality and responsiveness of customer support teams
4. Number and type of data sources covered
5. Costs and payment plans

Many publications offer aggregate reviews and ratings of data integration tools, as they do for data warehouses and business intelligence tools. Be sure to comparison-shop.

Make sure the data integration tools you are considering are compatible with the data warehouses and BI tools you have or are considering.

Calculate Total Cost of Ownership and ROI

The modern data stack promises substantial savings of time, money and labor. Compare your existing data integration workflow with a range of possible candidates.

Calculate the cost of your current data pipeline, which might require a careful audit of prior spending on data integration activities. You'll need to consider the sticker price, costs of configuration and maintenance, and any opportunity costs incurred by failures, stoppages and downtime. You should also consider the costs of your data warehouse and BI tool.

On the other side of the ledger, you will want to evaluate the benefits of the potential replacement. Some may not be very tangible or calculable (i.e., improvements in the morale of analysts), but others, such as time and money gains, can be readily quantified.

Establish Success Criteria

What should your analytics practice look like if you have successfully implemented a modern data stack?

Key criteria include:

1. Time, labor and monetary savings compared with the previous solution
2. Expanded capabilities of the data team
3. Successful execution of new data projects, such as customer attribution models
4. Reduced turnaround time for reports
5. Reduced data infrastructure downtime
6. Higher rates of business intelligence tool adoption within your organization
7. New metrics that are available and actionable

Set Up a Proof of Concept

Once you have narrowed your search to a few candidates and determined the standards for success, test the products out in a low-stakes manner. Most products will offer free trials for a few weeks at a time.

Set up connectors between your data sources and data warehouses, and measure how much time and effort it takes to sync your data. Perform some basic transformations. Set aside dedicated trial time for your team, and encourage them to stress-test the system in every way imaginable.

Compare the results of your trial against your standards for success.

> **REMEMBER:** *Automated data integration can radically enhance the capabilities of your analysts and data engineers, but you should fully understand your needs and what you intend to accomplish before you start. Develop a sense of what success (or failure) will look like, and make sure you stress-test a data stack before fully adopting it.*

Competitive analytics starts with automated data integration

The growth of cloud data has created unprecedented opportunities to develop new products and pursue advanced analytics. The volume and complexity of this data, however, pose an integration challenge that few organizations are equipped to handle. Automated data integration helps businesses fully harness data to accelerate and improve strategic decision-making. In this guide, we survey the history of data integration, evaluate current solutions, and show you how to choose the best integration tool for your business.

Inside:

- The evolution of the modern data stack
- Why data integration is even harder than it used to be
- How automation is crucial to modern data integration
- Building vs. buying a data integration solution
- Business and technical criteria for choosing a data integration solution
- How to implement a data integration solution

Charles Wang, Product Evangelist at Fivetran, has previously worked as a data analyst, data scientist and product manager.

About Fivetran

Fivetran is the leader in automated data integration, delivering ready-to-use data connectors, transformations and analytics templates. Data connectors by Fivetran continuously supply data to a central repository, and adapt as schemas and APIs change. This ensures effortless and reliable data access, empowering you to adopt as many SaaS apps as you need and pursue analytics with confidence. Learn more about the future of data integration at fivetran.com.

ISBN 978-1-7346299-0-3

90000>

9 781734 629903