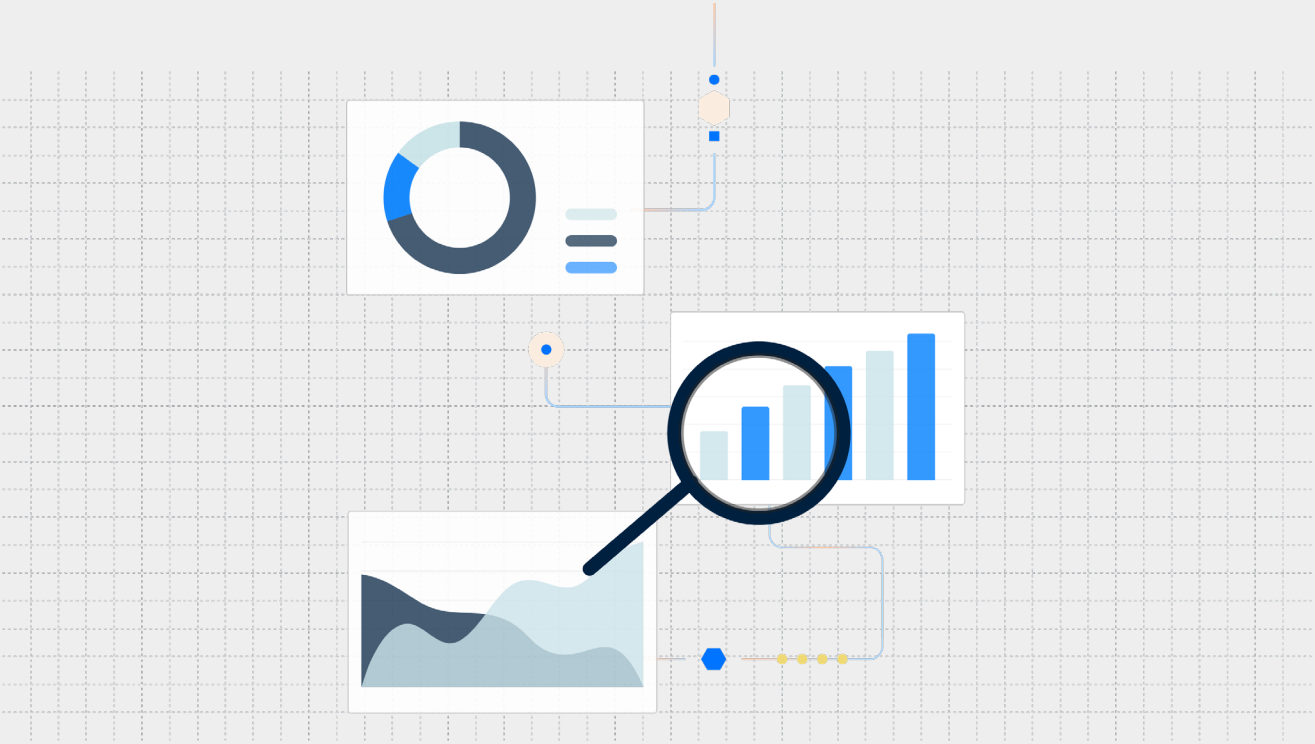
The ultimate guide to data integration

Mental models for data integration, analytics and the modern data stack



Fivetran

Table of contents

Introduction

The world is awash with data. Almost every activity leaves behind a digital footprint that is an element of a much bigger picture and a clue to deeper insight. Data analysis is critical to every industry, helping business leaders make smarter decisions about what products to deliver, market segments to target, logistical arrangements to make and more. With the rise of cloud computing, SaaS technology and the **modern data stack**, insights that were previously only accessible to the deepest pockets and the largest teams are now accessible to organizations of all sizes and means. The time is ripe for a new approach and mindset toward analytics and data integration.

Without data integration, data – and the information, knowledge and insight that can be gleaned from it – is siloed. Siloed data hampers collaboration by producing partial and conflicting interpretations of reality, making it difficult to establish a single source of truth and bring everyone in an organization to the same page.

True to its name, data integration is also key to data integrity – ensuring the accuracy, completeness and consistency of data. Data integrity is a matter of fully comprehending data, having confidence in its completeness and ensuring its accessibility where needed.

If you are an analyst, data engineer or data scientist (or manager of the aforementioned) at an organization that uses operational systems, applications and other tools that produce digital data, this guide is for you. Your role should allow you to influence or determine the tools your company uses. More importantly, it should put you in a position to influence how teams think about solving problems, finding insights and making decisions with data. This guide is as much about building a modern data mindset as it is about implementing new tools and processes.

With the modern data mindset, you will be able to leverage automation into substantial savings of time, talent and money as well as the opportunity to pursue higher-value uses of data. By contrast, as long as your organization remains tethered to outmoded methods and mindsets of data integration, you will leave money on the table, waste the efforts of your data professionals and lose ground to savvier entrants in the marketplace.

We'll explain and evaluate the various approaches to data integration currently available and you'll learn how to bring data integration to your organization. Ultimately, we'll build a modern data mindset and equip you to coach others to adopt the same.

Chapter 1: Why data matters

Data integration refers to the processes used to manage and combine flows of data from various sources. With all of your organization's data in a single environment, you can construct a comprehensive view of your business operations. A common example is examining customer journeys by tracing how customers interact with marketing, sales, product, customer support and more. This practice of using data to solve problems is analytics.

Typical business use cases for analytics include mining data for insights to improve the following:

1. Customer experiences
2. Internal processes and operations
3. Products, features and services

Data can also be an essential building block of machine learning or artificial intelligence products. It can also *be* a product when packaged in a way that is legible and helpful to customers.

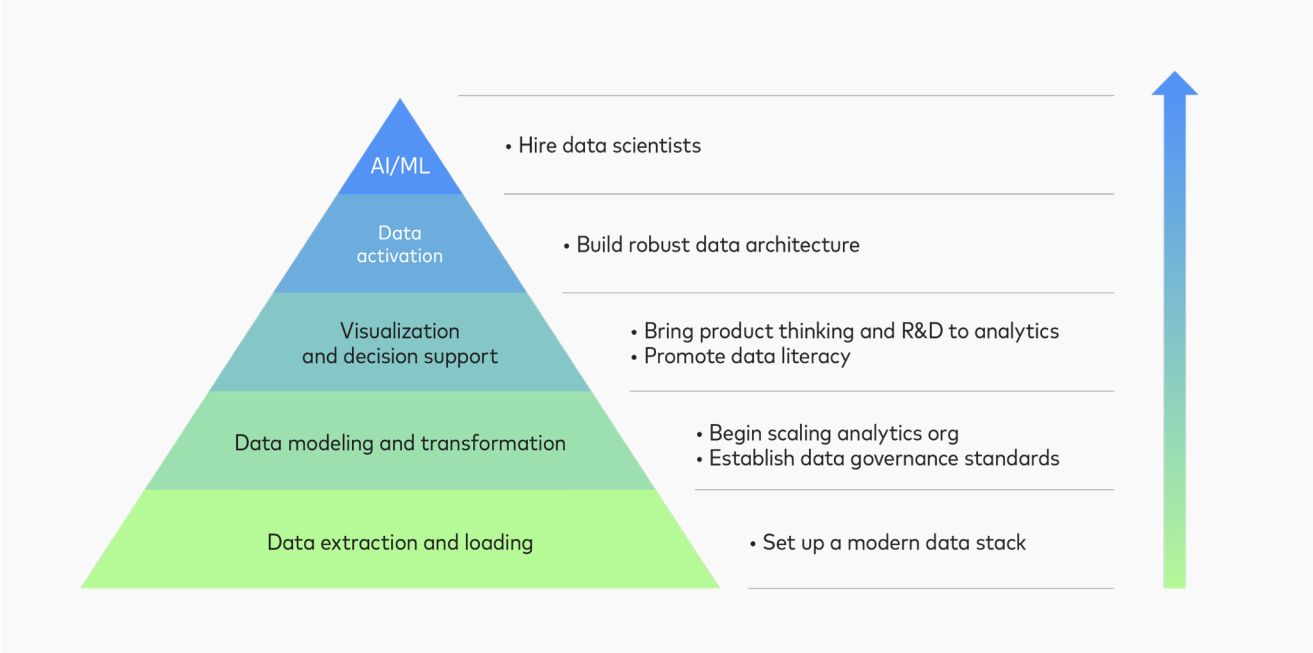
As your data integration and analytics capabilities mature, your organization will be able to promote widespread data literacy and become more nimble, dynamic and innovative.

The data hierarchy of needs

Building a modern approach to data integration and analytics requires a solid foundation. We can place data integration and analytics in a **hierarchy of needs** with corresponding steps to fulfill those needs:

1. **Data extraction and loading.** This is the first step to data integration. You must gather data and make it available on a single platform. This is best accomplished with the help of a suite of tools called the modern data stack.
2. **Data modeling and transformation.** Once data from multiple data sources is in one place, your analysts can start massaging it into structures that can support dashboards, visualizations, reports and predictive models. As your needs grow over time, this will require scaling your data team and establishing standards for data governance.

3. **Visualization and decision support.** Now, you are ready for analytics. Your data models will enable you to create a comprehensive view of your operations. Build reports and dashboards as needed. To best accommodate these activities, you will need to bring product management best practices into the production of data assets and promote data literacy across your organization.
4. **Data activation.** Analytics data can be routed back into your operational systems to give your team members real-time visibility into operations or to automate business processes that depend on specific inputs of data. Bringing data directly into production systems can require specialized tools or data engineering expertise.
5. **AI and machine learning.** The pinnacle of data science is building systems that leverage predictive modeling. There are countless applications of machine learning, ranging from making predictions on the basis of linear regression to self-driving vehicles. At this stage, your organization should begin hiring specialists such as data scientists and machine learning engineers.



The business benefits of analytics

Analytics does more than surface interesting information and insights. It goes beyond academic understanding of a situation to inform practical business decisions, bringing real benefits to every organizational team or function.

Traditionally, analytics was used for basic problem solving. This practice continues today, but the tools, technologies and teams performing this work are increasingly characterized by massive scale and predictive modeling, as exemplified by artificial intelligence and machine learning.

Automation and computation do more than just increase speed and efficiency or save time, talent and money. The ability to crunch more data in minutes than any human could in a lifetime and reach insightful conclusions that would otherwise be impossible marks a qualitative difference in capability. Thinking about data analytics this way is a shift toward the modern data mindset, where improvements to process and decision-making capacity are heavily informed by advances in technology, enabled by a modern data stack.

Common everyday applications of artificial intelligence and machine learning include:

1. **Recommendations** – Personalized experiences in advertising, social media feeds, streaming services and reviews.
2. **Anomaly detection** – Fraud detection, image recognition and medical diagnostics.
3. **Numerical predictions** – Examining causal relationships, such as crop yields as a function of fertilizer use, blood pressure as a function of medication or revenue as a function of advertising spending.
4. **Intelligent agents** – Building real or virtual machines capable of independent decision-making, such as chatbots, self-driving vehicles and automated production lines.

Chapter 2: What is data integration?

Data integration encompasses the extraction, loading and transformation of data into usable data models. Data integration enables analytics by combining data from across an organization in one platform and modeling it into a representation of reality that decision-makers can easily interpret.

Data can be produced from a wide range of events or activities:

1. **Manual data entry**, such as survey forms collected and processed by an office
2. **Sensor inputs**, such as scans at a checkout line
3. **Digital documents, content and media**, such as social media posts
4. **Digital activity** recorded by software triggers, such as clicks on a website or app

Before anyone can use data to recognize patterns and identify causal mechanisms, the data must be in one place and organized in a manageable fashion.

The data integration process

In order to make raw data ready for analytics, several things must happen:

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 destination.
3. Data is cleansed and modeled to meet the analytics needs of various business units.
4. Data is used to power products or business intelligence.

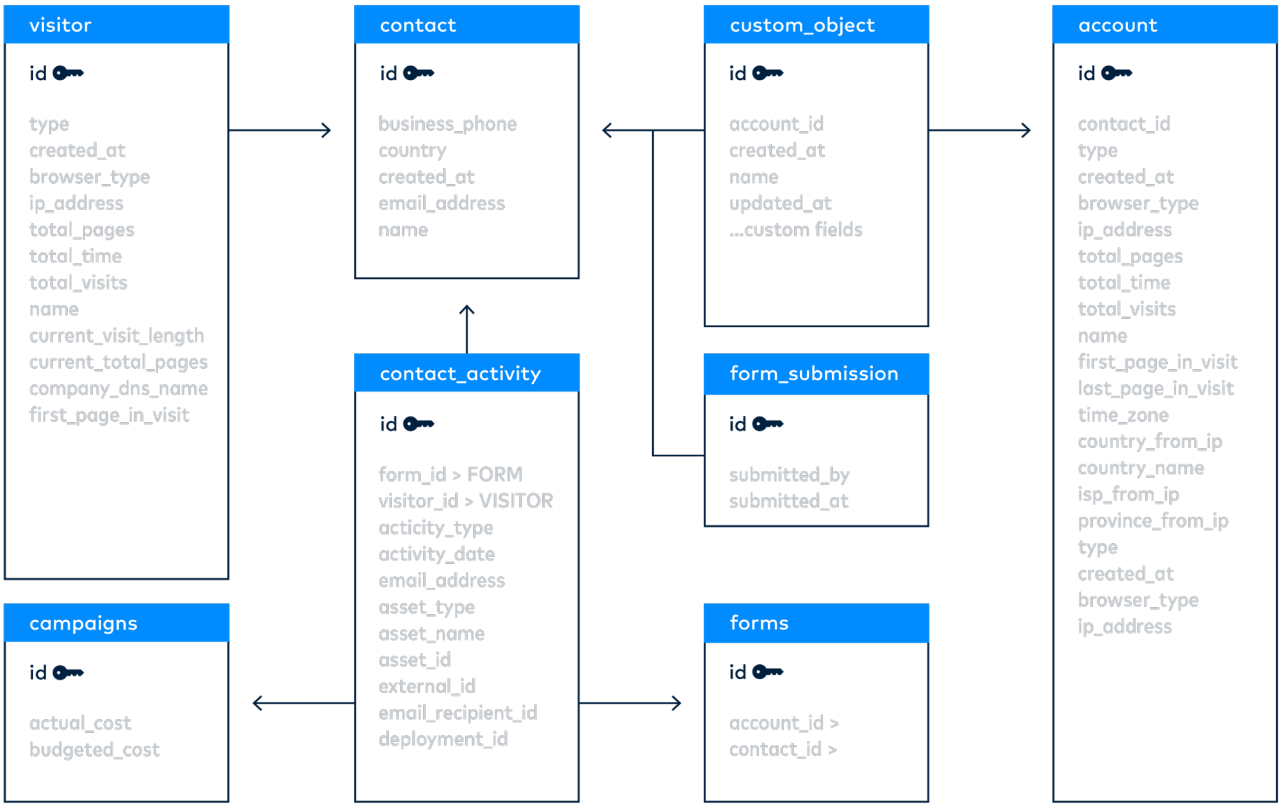
Data integration can be performed in a manual, ad hoc manner or programmatically using software. The ad hoc approach is slow, error-prone, unreplicable and unscalable. It is costly as well, demanding the attention of skilled, highly-paid data professionals, yet **62% of organizations** routinely use spreadsheets to manually combine and visualize data. Many data scientists use the method to produce ad hoc reports.

By contrast, a modern data mindset opens up far better ways of working. The programmatic approach to data integration involves a suite of tools called a modern data stack. The data integration software used in a modern data stack can be written in-house by an organization's engineering team or wholly outsourced and automated. This systematic approach is faster, more reliable, more accurate and more cost effective than manual integration. And it makes data engineers, analysts and data scientists happier too, since their time is spent on meaningful work rather than slow, repetitive data collection and cleaning.

Important characteristics of data sources and destinations

Data sources include files, databases and API endpoints. In many cases, these are produced by applications. Every data source has an underlying data model that reflects some version of reality. Data models for applications, for instance, illustrate how users interact with the product. A **data model** is an abstract, strictly formatted representation of reality. A **schema** is a practical blueprint for turning a data model into a database with tables, rows, columns and interrelations.

Standardized schema



The destination in which data is centralized to become a source of truth is typically a relational database that is optimized for analytics called a **data warehouse**. Data warehouses store structured data that follows highly specific formatting rules for the sake of easy interpretation by machine and is organized into tables with rows and columns.

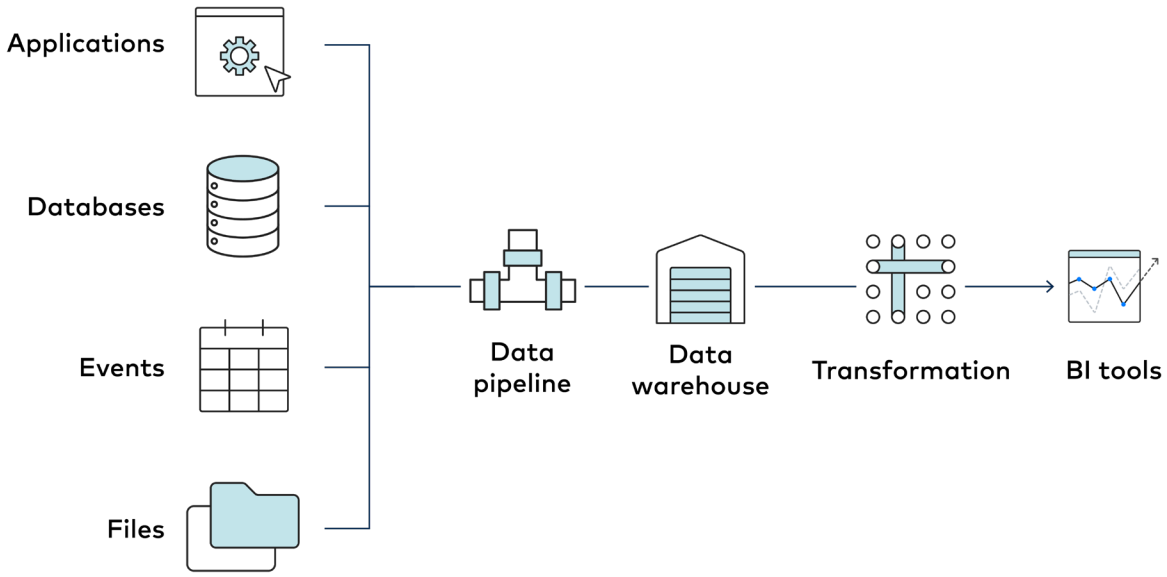
In some cases, a destination might instead be a **data lake**. A deeper discussion of the tradeoffs between data lakes and data warehouses is beyond our scope here, but in short, data lakes are best used to support use cases that require unstructured data and mass storage of media files or documents, such as certain forms of machine learning. Newer technologies combine the functionality of both data warehouses and data lakes under a common cloud data platform, an integrated solution that supports analytics, machine learning, data replication, data activation and all other data needs.

What is a modern data stack?

We've already touched on the modern data stack a few times, but let's dig into the detail: A data stack is a suite of tools and technologies used for programmatic data integration. The modern data stack leverages new technological developments in the cloud and automation. Its basic elements include:

1. **Data sources** – These typically include:
    a. Applications
    b. Databases
    c. Files
    d. Digital events
2. **Data pipeline** – Software featuring data connectors that extract data from a source and load it into a destination. Data connectors may also apply light transformations such as normalizing and cleaning data, orchestrate transformations into models for analysts or simply load raw data.
3. **Destinations** – Central data repositories, typically either **data lakes or data warehouses**, that permanently store large amounts of data.
4. **Transformation and modeling layer** – It is usually necessary to transform raw data to ready it for analysis. This can involve **joining tables together, performing aggregate calculations, pivoting data or reformatting data**. Transformations may be performed in staging environments within the data pipeline (ETL) or within the data warehouse (ELT).

5. **Analytics tools** – These include off-the-shelf business intelligence platforms for reporting and dashboards, as well as analytics and data science packages for common programming languages. Analytics tools are used to produce visualizations, summaries, reports and dashboards.



You can think of data integration as a specific type of a more general operation called data movement. Aside from data integration, data movement also includes data replication between operational systems for the sake of redundancy and performance. Data activation pushes data that has been modeled for analytics from a destination back into operational systems, enabling all manner of data-driven business process automation.

Chapter 3: Approaches to data integration

There are two main approaches to data integration. One of these architectures, Extract, Transform, Load (ETL), is widespread, but costly and rapidly becoming obsolete. The other, Extract, Load, Transform (ELT), is far more accessible and leverages continuing advancements in technology.

Before we dig into the differences in approach, let's understand the actions involved.

Transformation explained

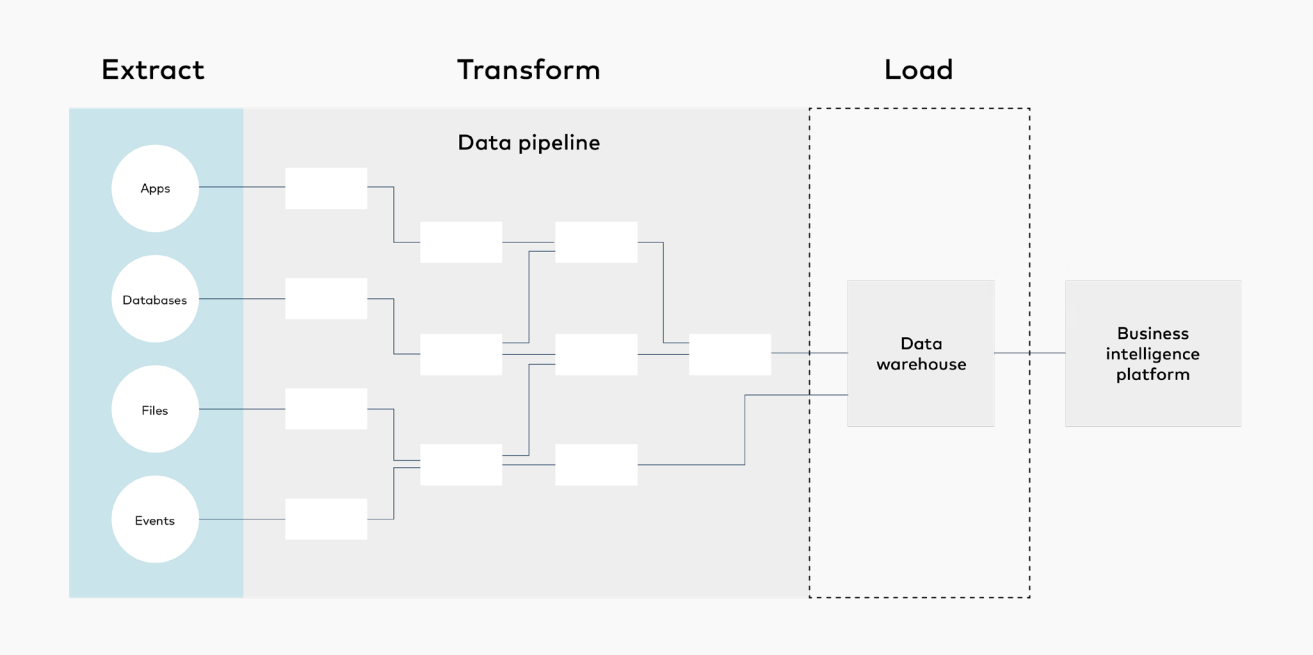
Transformation plays a central role in both **ETL and ELT**. We can define transformation as any of the operations involved in turning raw data into analysis-ready data models. These include:

1. **Revising** data ensures that values are correct and organized in a manner that supports their intended use. Examples include fixing spelling mistakes, changing formats, hashing keys, deduplicating records and all kinds of corrections.
2. **Computing** involves calculating rates, proportions, summary statistics and other important figures, as well as turning unstructured data into structured data that can be interpreted by an algorithm.
3. **Separating** consists of dividing values into their constituent parts. Data values are often combined within the same field because of idiosyncrasies in data collection, but may need to be separated to perform more granular analysis. For instance, you may have an "address" field from which street address, city and state need to all be separated.
4. **Combining** records from across different tables and sources is essential to build a full picture of an organization's activities.

Transformation can be complicated and computationally intensive, depending on the careful sequencing and orchestration of different operations. It is highly sensitive to schema changes, both as upstream data sources evolve and downstream business needs change. This has critical implications for how scalable and adaptable a data integration architecture can be. Where transformation takes place in the data integration workflow can make a huge difference in terms of the time, talent and money involved in building and maintaining data pipelines.

What is ETL?

The traditional approach to data integration, Extract, Transform, Load (ETL), dates from the 1970s and is so ubiquitous that "ETL" is often used interchangeably with data integration. Under ETL, data pipelines extract data from sources, transform data into data models for reports and dashboards and then load data into a data warehouse.

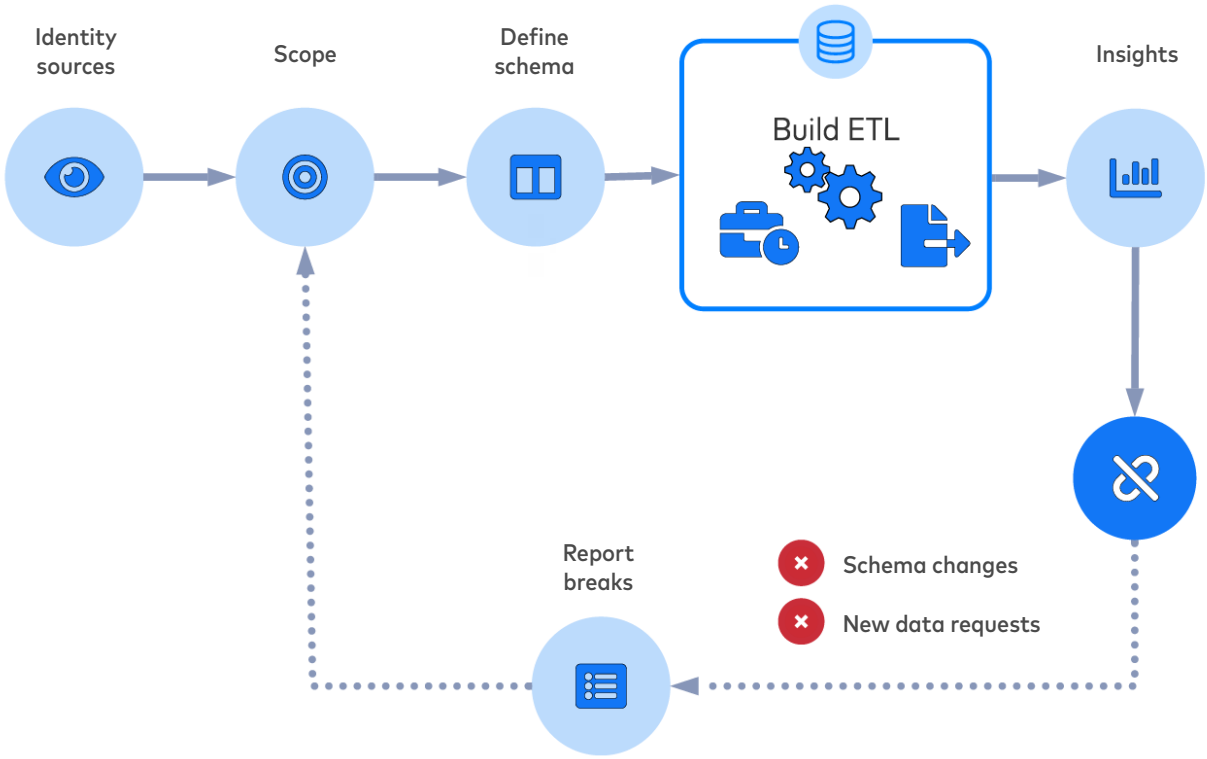


In ETL, data transformations typically aggregate or summarize data, shrinking its overall volume. By transforming data before loading it, ETL limits the volume of data that is warehoused, preserving storage, computation and bandwidth resources. When ETL was invented in the 1970s, organizations operated under an extreme scarcity of storage, computation and bandwidth. Constrained by the technology of the time, they had no choice but to spend valuable engineering time constructing bespoke ETL systems to move data from source to destination.

The project workflow for ETL consists of the following steps:

1. Identify desired data sources.
2. Scope the exact analytics needs the project solves.
3. Define the data model/schema that the analysts and other end-users need.
4. Build the pipeline, consisting of extraction, transformation and loading functions. This requires a significant outlay of engineering time.
5. Analyze the data to extract insights.

ETL workflow



Under ETL, extraction and transformation are tightly coupled because both are performed before any data is loaded to a destination. Moreover, because transformations are dictated by an organization's specific analytics needs, every ETL pipeline is a complicated, custom-built solution. The bespoke nature of ETL pipelines makes it especially challenging to add or revise data sources and data models.

You're probably beginning to see the flaws and potential pitfalls in this approach. It was necessary when computing and storage resources were limited, and acceptable while data sources remained constant and predictable.

But the world has changed. The ETL workflow - including all the effort to scope, build and test it - must be repeated whenever upstream schemas or data sources change – specifically, when fields are added, deleted or changed at the source - or when downstream analytics needs change, requiring new data models.

Imagine an application rearranges the tables in its data model in order to support new customer data. The pipeline code for extracting, transforming and loading the data depends on the old schema and will no longer work and will have to be rewritten. This stoppage will halt all updates, preventing a business from making up-to-date, informed decisions.

In the second case, an analyst might want to create a new attribution model that requires several data sources to be joined together in a new way. As with the previous case, this means a slow, labor-intensive process of rebuilding the workflow.

Since extraction and transformation precede loading, all transformation stoppages prevent data from being loaded to the destination, causing data pipeline downtime.

ETL for data integration therefore involves the following challenges:

- **Perpetual maintenance and revision** – Since the data pipeline both extracts and transforms data, the moment upstream schemas or downstream data models change, the pipeline breaks and the code base must be revised.
- **Customization and complexity** – Data pipelines perform sophisticated transformations tailored to the specific analytics needs of the end users. This means custom code.
- **Lost engineering resources** – The system requires the full-time efforts of engineers to build and maintain because it runs on a bespoke code base.

These difficulties are further aggravated if the ETL setup is on-premise, hosted in data centers and server farms directly run by an organization. These setups require yet more tuning and hardware configuration.

The key tradeoff made by ETL is to conserve computation and storage resources at the expense of labor. This made sense when data needs were much simpler and computation and storage were extremely scarce, especially in comparison to labor. These circumstances are no longer true, and ETL in the present day is a hugely expensive and labor-intensive undertaking with dubious benefits.

Game-changing technology trends make ETL obsolete

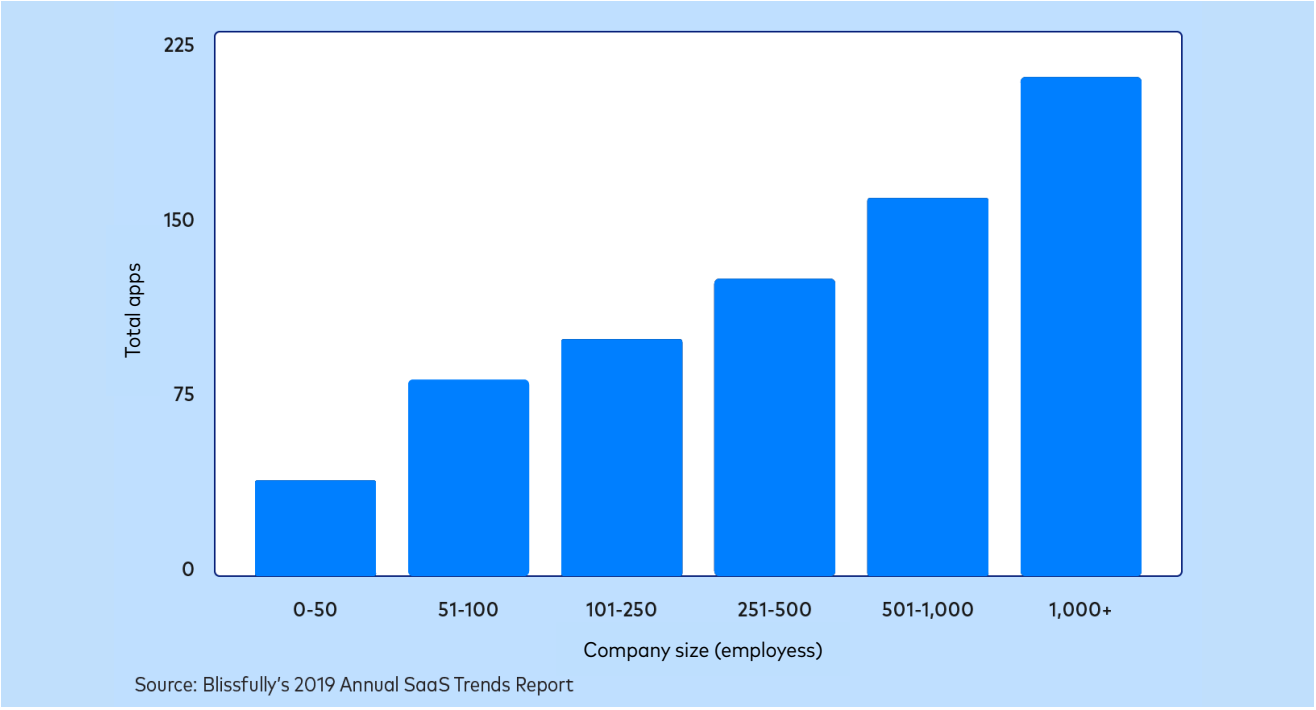
In recent years, "big data" and the "cloud" have entered common use as buzzwords, and they both relate directly to the demise of ETL as a practical method for data integration and analysis. Big data refers to the massive volume and complexity of modern data and arises from the growth of internet-based, decentralized computation and storage, also known as the cloud. Cloud-based applications and devices (including the growing **internet of things**) produce extensive digital footprints of data that can be turned into valuable insights.

This data is typically stored in cloud-based files and operational databases, and then exposed to end users as:

1. API feeds
2. File systems
3. Database logs and query results
4. Event streams

Cloud-based technologies, especially Software-as-a-Service (SaaS), have become increasingly prolific. A typical organization now uses dozens or hundreds of applications.

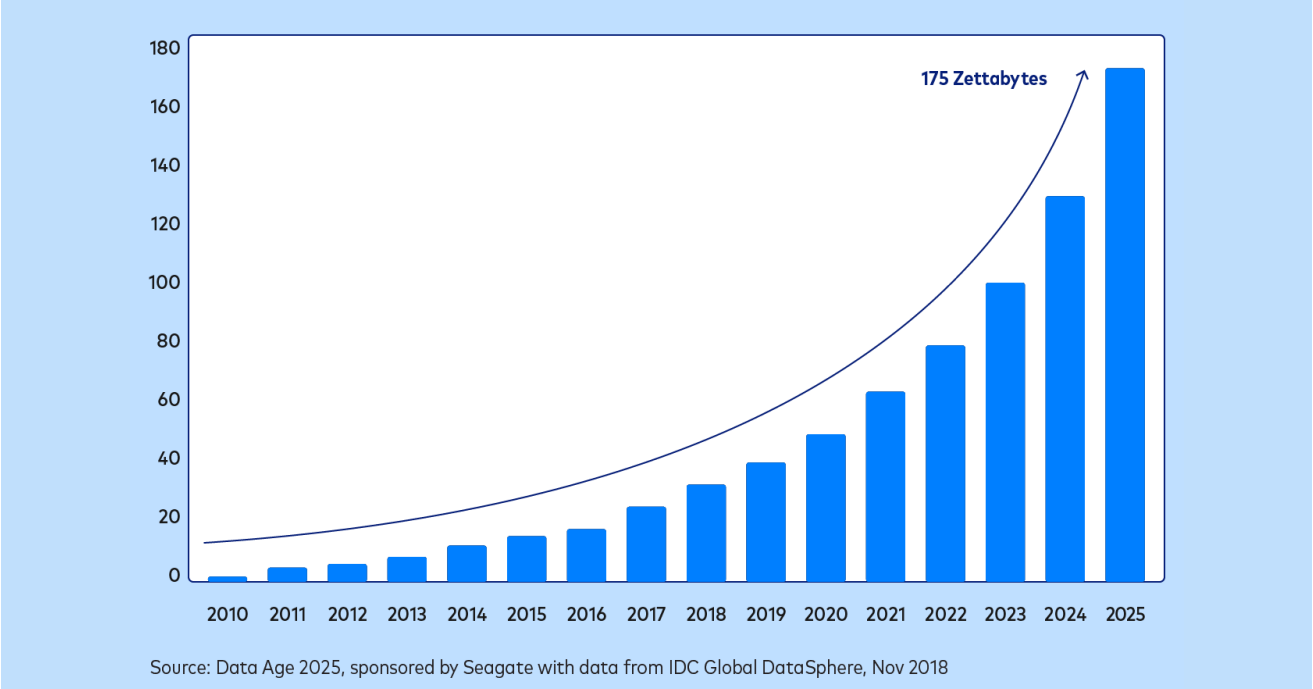
Number of apps per company



Source: Blissfully's 2019 Annual SaaS Trends Report

As a result, the overall volume of data produced worldwide has exploded in recent decades:

Annual size of the global datasphere

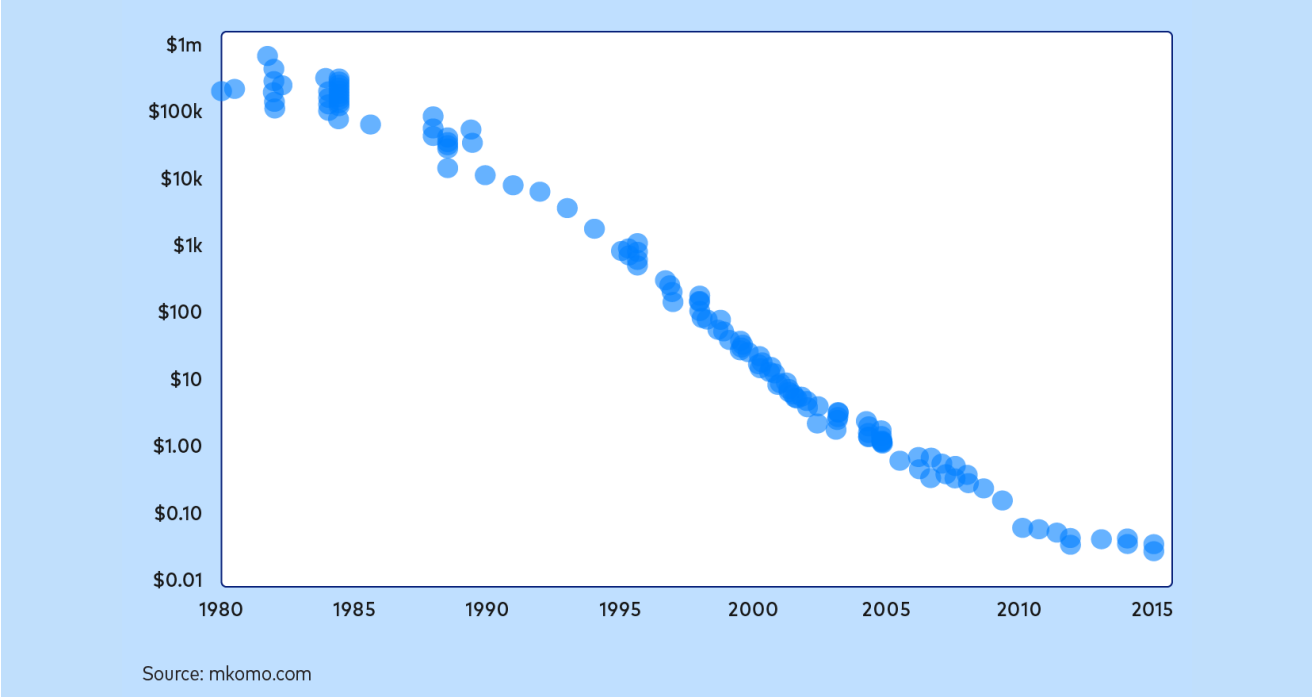


Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018

As the volume and granularity of data continue to grow, so do opportunities for analytics. More data presents the opportunity to understand and predict phenomena with greater accuracy than ever.

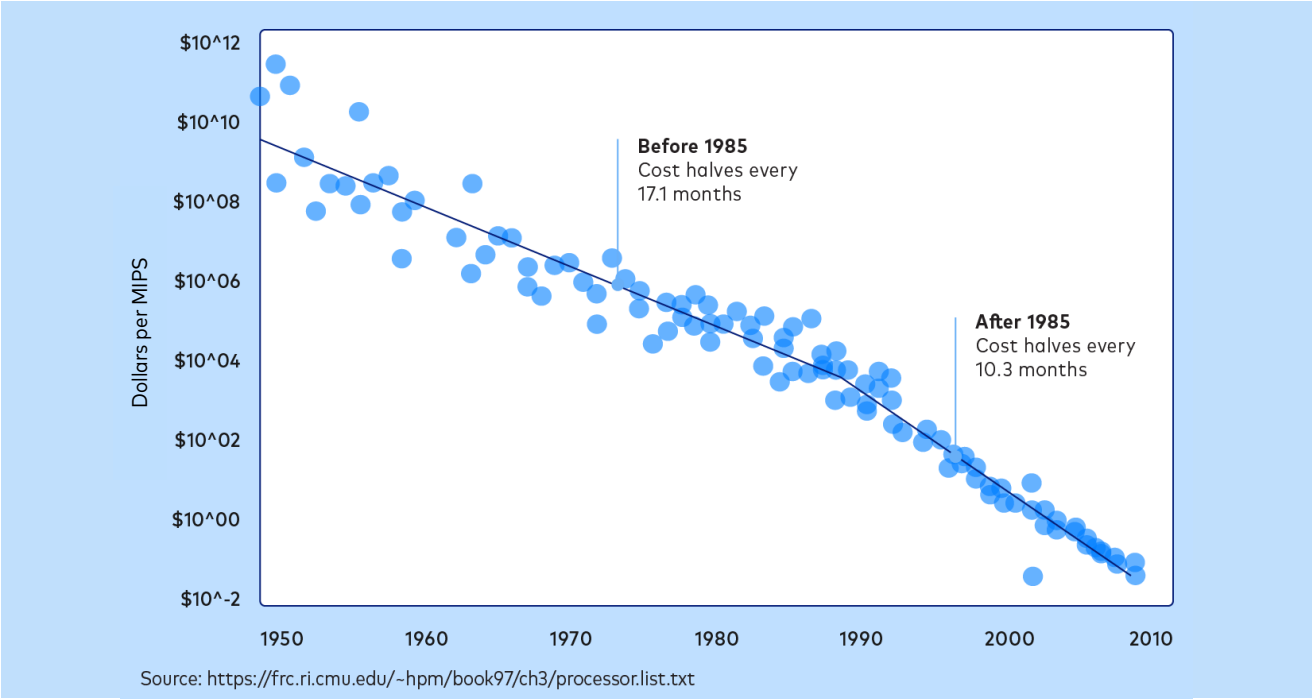
ETL made sense at a time when computation, storage and bandwidth were extremely scarce and expensive. These technological constraints have since disappeared. The cost of storage has plummeted from nearly $1 million to a matter of cents per gigabyte over the course of four decades:

Hard drive cost per gigabyte



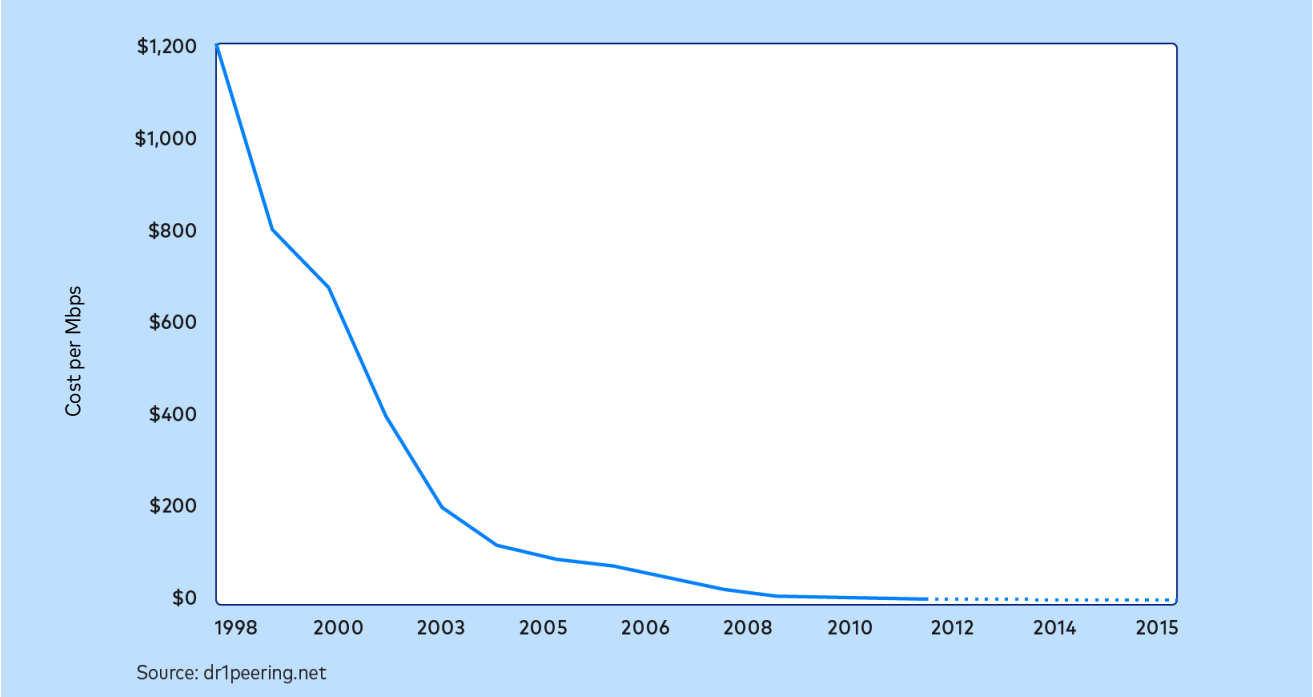
The cost of computation has reduced by a factor of millions since the 1970s:

Cost of computing



And the cost of internet transit has fallen by a factor of thousands:

Price of internet traffic (bandwidth)



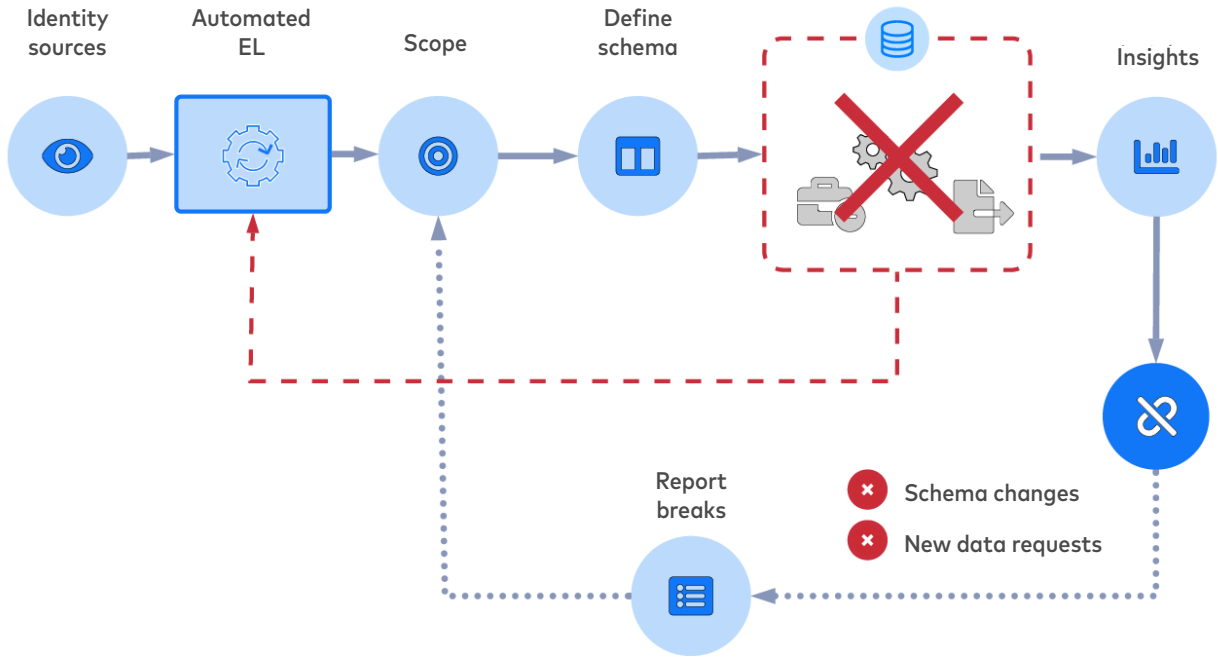
These trends have made ETL and the costly, labor-intensive efforts necessary to support it obsolete in two ways. First, the affordability of computation, storage and internet bandwidth has led to the explosive growth of the cloud and cloud-based services. As the cloud has grown, the volume, variety and complexity of data have grown as well. A brittle pipeline that integrates a limited volume and granularity of data is no longer sufficient.

Secondly, the affordability of computation, storage and internet bandwidth allows modern data integration technologies to be based in the cloud and store large volumes of untransformed data in data warehouses. This makes it practical to reorder the data integration workflow, in the process saving considerable money and headcount.

ELT: A modern alternative to ETL

The ability to store huge quantities of untransformed data in data warehouses enables a new data integration architecture, Extract, Load, Transform (ELT), in which transformation takes place at the end of the workflow and data is more-or-less immediately loaded to a destination upon extraction.

ETL workflow

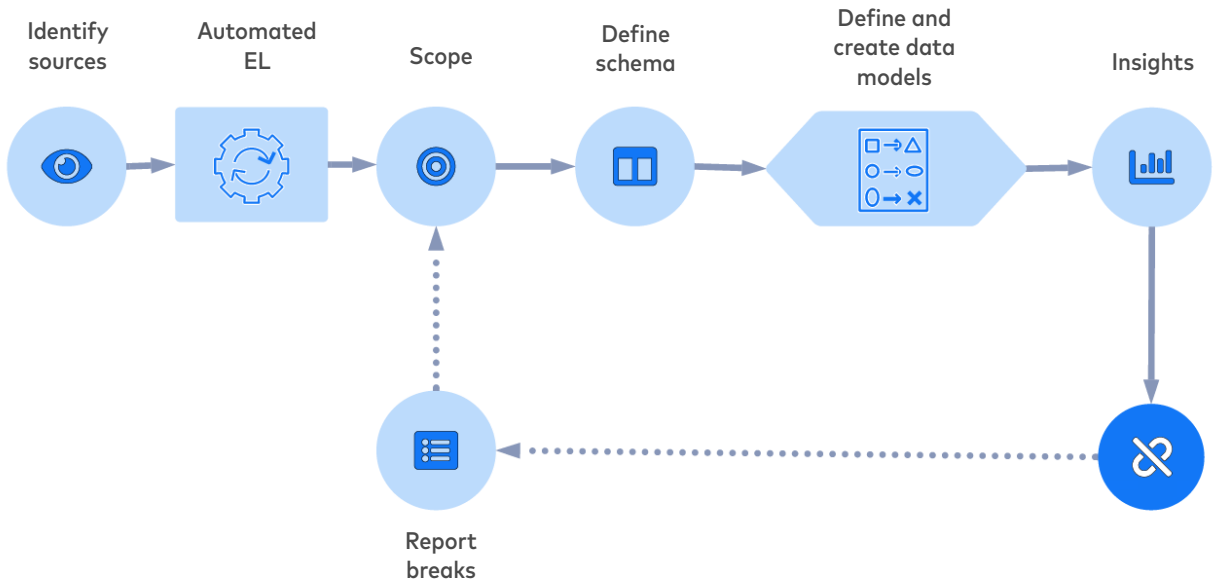


This prevents the two failure states of ETL (i.e. changing upstream schemas and downstream data models) from interfering with extraction and loading, leading to a simpler and more robust approach to data integration.

The ELT workflow features a shorter project cycle than ETL:

1. Identify desired data sources
2. Perform automated extraction and loading
3. Scope the exact analytics needs the project is meant to solve
4. Create data models by building transformations
5. Perform analytics and extract insights

ELT workflow

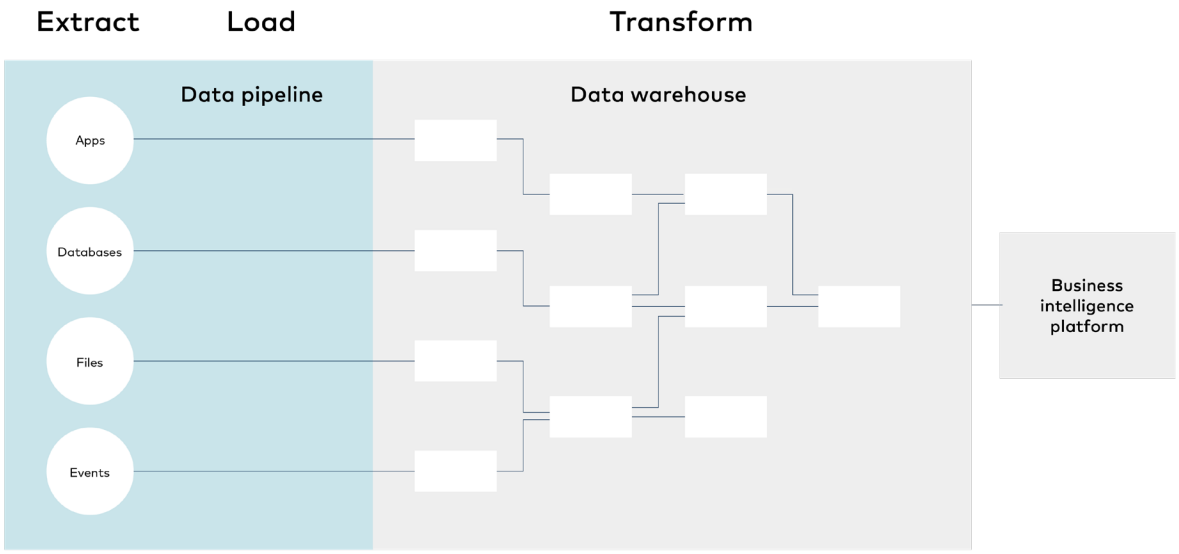


Under ELT, extracting and loading data are upstream, and therefore independent, of transformation. Although the transformation layer may still fail as upstream schemas or downstream data models change, these failures do not prevent data from being loaded into a destination. An organization can continue to extract and load data even as transformations are periodically rewritten by analysts. Since the data is warehoused with minimal alteration, it serves as a comprehensive, up-to-date source of truth.

Moreover, since transformations are performed within the data warehouse environment, there is no longer any need to arrange transformations through drag-and-drop transformation interfaces, write transformations as Python scripts or build complex orchestrations between disparate data sources. Instead, transformations can be written in SQL, the native language of most analysts. This turns data integration from an IT- or engineer-centric activity featuring long project cycles and the heavy involvement of engineers to more of a self-service activity directly owned by analysts.

A further benefit of performing transformations in the data warehouse environment is performance. Cloud-based data warehouses can scale additional compute and storage as needed, enabling easier and faster transformations compared with the layers of staging environments used in ETL. The cloud enables just-in-time provisioning of resources, sparing the expense of maintaining excess hardware capacity that is only occasionally used.

Finally, and most crucially, the ELT architecture makes extraction and loading practical to outsource, automate and standardize. Since transformations are performed within the warehouse, the "EL" part of the pipeline does not need to produce a differentiated output based on an organization's specific needs.



Instead, an external provider such as Fivetran can provide a standardized schema to every customer. Since there are relatively **few ways to normalize** a schema, the most sensible way to standardize a data model is through **normalization**. Normalization fosters data integrity, ensuring that data is accurate, complete, consistent and of known provenance. It also makes data models easier for analysts to interpret. Standardized outputs have the added benefit of enabling derivative products such as standardized **data model templates** for analytics.

In short, ELT turns data models into undifferentiated commodities, enabling outsourcing and automation. It empowers a data team to shift in focus from building, maintaining and revising a complicated piece of software to simply using its outputs and building data models directly in the destination.

Major differences between ETL and ELT

The following table summarizes the differences between ETL and ELT:

ETL	ELT
Extract, transform, load	Extract, load, transform
Integrate summarized or subsetted data	Integrate all raw data
Loading and transformation tightly coupled	Loading and transformation decoupled
Longer time to load data	Shorter time to load data
Transformation failures stop pipeline	Transformation failures do not stop pipeline
Predict user-cases and design data models beforehand or else fully revise data pipeline	Create new use-cases and design data models any time
Bespoke	Off-the-shelf
Constant building and maintenance	Automated
Conserves computation and storage	Conserves labor
Use scripting languages for transformations	Use SQL for transformations
Engineering/IT-centric; expert system	Analyst-centric; accessible to non-technical users
Cloud-based or on-premise	Almost strictly cloud-based

There are some cases where ETL may still be preferable over ELT. These specifically include cases where:

1. The desired data models are well-understood and unlikely to change quickly. This is especially the case when an organization also builds and maintains the systems it uses as data sources.
2. There are stringent security and regulatory compliance requirements concerning the data and it cannot be stored in any third-party location, such as cloud storage.

These conditions tend to be characteristic of very large enterprises, those in highly-regulated industries and organizations that specialize in Software-as-a-Service products. In such cases, it may still make sense to use automated, outsourced ELT to integrate data from third-party data sources while building and maintaining ETL to integrate in-house, proprietary data sources.

The benefits of ELT and automation

An organization that combines automation with ELT can dramatically simplify its data integration workflow. Automated ELT is a force multiplier to data engineering, enabling teams to focus on more mission-critical projects such as optimizing an organization's data infrastructure or bringing data models into production rather than constructing and maintaining data pipelines. By performing the work of several full-time engineers, automated ELT allows teams to remain lean and use their headcount much more efficiently.

A further benefit of automation over manual data integration is repeatability and consistency of quality. Analysts and data scientists can finally use their understanding of business to model and analyze data **instead of wrangling or munging** it.

Chapter 4: Build vs. buy for data integration

There are tradeoffs between building your own data integration and buying a solution as a service but, in most cases, it makes more sense to buy an off-the-shelf solution.

The cost of building a data pipeline

When **Wakefield Research** surveyed large (2,500+ headcount) companies in the US, UK, Germany and France they found a nominal average yearly cost of nearly $520,000 to build and maintain data pipelines. At an average cost of $98k per year each, this amounts to the work of more than five full-time data engineers.

Here is another calculation for a simpler use case. You can follow along below, or **use our calculator**.

In order to estimate the cost of building data pipelines, we need the following:

1. Average yearly cost of labor for your data engineers (or analysts or data scientists)
2. How many data sources you have
3. How long it takes to build and maintain a typical data source

With these figures, we can estimate the time and money spent on engineering.

- Let's assume the cost is $140,000 for each data engineer: $100,000 base salary, with a 1.4 multiplier for benefits

Assume that it takes about 7 weeks to build a connector and about 2 weeks per year to update and maintain it. Each connector takes about 9 weeks of work per year.

- With seven connectors, that's 7 * (7 + 2) = 63 weeks of work per year

Use the weeks of work per year to calculate the fraction of a working year this accounts for. Then, multiply it by the cost of labor to arrive at the total monetary cost. Assume the work year lasts 48 weeks once you account for vacations, paid leave and other downtime.

- If the cost of labor is $140,000, seven connectors take 63 weeks of work, and there are 48 working weeks in a year, then ($140,000) * (63 / 48) = $183,750

Based on our experiences at Fivetran, these figures are realistic estimates for understanding the cost of a DIY data integration solution. On average, we find that our customers save the equivalent of roughly two engineers' salaries. At a minimum, it is likely to be a high five-figure or low six-figure commitment, even for a relatively simple use case involving a small number of data sources.

More importantly, there are serious opportunity costs to using engineering time for building and maintaining data pipelines. There are many high value data projects that engineers, analysts and data scientists can build downstream of data pipelines, such as:

1. Data models
2. Visualizations, dashboards and reports
3. Customer-facing production systems
4. Business process automations
5. API feeds and other data products for third parties
6. Custom data pipelines to sources for which off-the-shelf pipelines don't exist
7. Infrastructure to support data activation
8. Predictive models, artificial intelligence and machine learning, both for internal use in for customer-facing products

Avoiding the integration iceberg: The technical risks of building a data pipeline

Designing a data pipeline is like looking at an iceberg. On the surface, it's simple. Data just needs to move from one location to another.

But much like an iceberg, the majority of the work involved is hidden from view, lurking beneath the surface. Data integration is complicated, requiring a complex system to solve a wide range of technical problems in the following areas:

- Automation
- Performance
- Reliability
- Scalability
- Security

Automation

The purpose of building a data pipeline is to programmatically move data from a source to a destination instead of manually wrangling files. In order to do this, your data engineers need to design and build a system that enables users to easily set the system to extract, transform and load data on a regular schedule without manual configuration and triggering.

Building this from top to bottom also means organizing raw data from the source into a structure that analysts can use to produce dashboards and reports. This requires a deep understanding of the underlying data.

Finally, you need to create tools to monitor the health of a pipeline, so that your teams are alerted the instant a problem arises, and troubleshoot any errors and stoppages that emerge, bringing us to the next points.

Performance

A good data pipeline must perform well and deliver data before it becomes too stale to be actionable. It must also avoid interfering with critical business operations while running.

One approach is to incrementally capture updates using logs, time stamps or triggers instead of querying entire production tables or schemas. A commonly-used term for this approach, especially when reading from databases, is change data capture.

Other performance-related considerations include parallelizing and distributing the architecture of the data pipeline to use more resources as needed - enabling the system to cope with increased load or demand. Scaling computation and storage resources up and down and properly budgeting for such activities can be challenging. Another consideration is to identify and mitigate performance bottlenecks by compartmentalizing and buffering sensitive operations.

Reliability

Performance is not very meaningful without consideration for reliability. There are many reasons a synchronization of data from one location to another may fail. One common reason is simply that schemas upstream change. Depending on how the data pipeline is architected, new or deleted tables and columns can completely derail the operation of the data pipeline.

Bugs and hardware failures of various kinds are also common. Your pipeline could experience memory leaks, network outages, failed queries and more. In order to recover from failed syncs, you will need to build an **idempotent** system, in which operations can be repeated to produce the same outcome after the initial application.

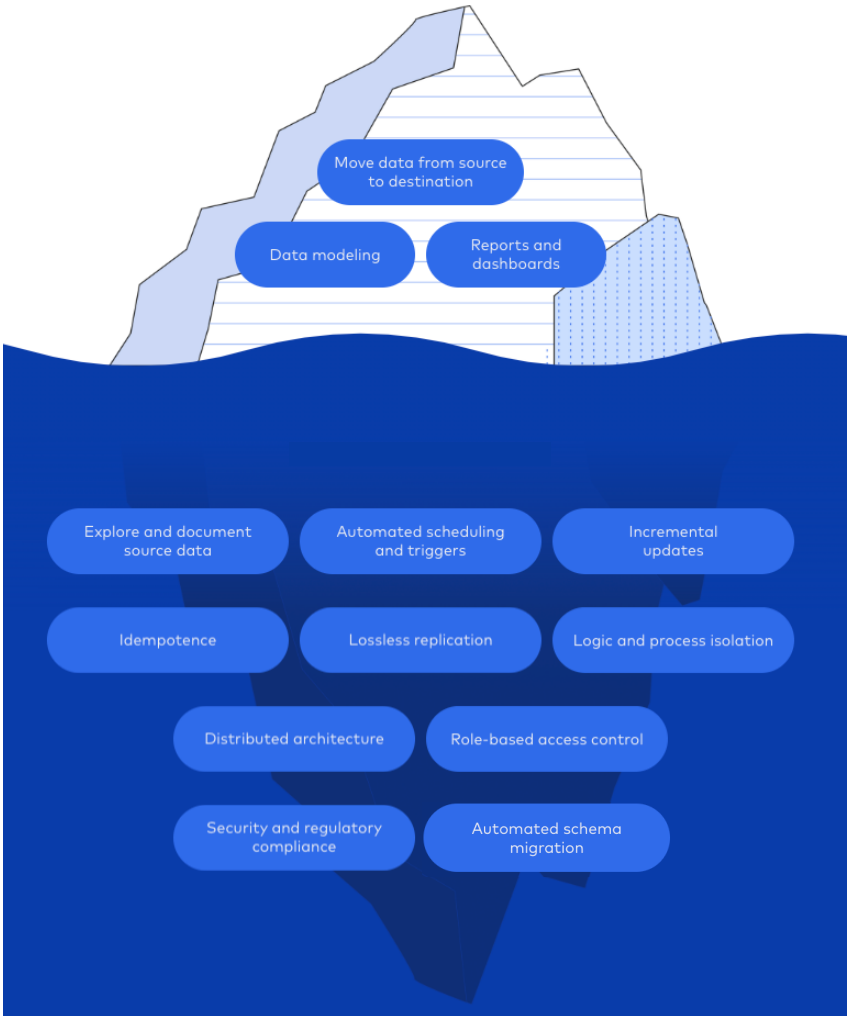
Scalability

As your organization grows, it will need to accommodate an increasing number of data sources and a higher volume of data. There is a good chance that you will also face more stringent performance requirements, such as shorter turnaround times, as your organization's data use matures. This will multiply your engineering burden considerably, as you will need to build and maintain connectors for each new data source.

As the number of your connectors and users grow, you may also eventually need to design a system to programmatically control your data pipeline.

Security

For the sake of regulatory compliance in many jurisdictions, your data pipeline shouldn't expose or store personally identifiable information. All traffic across your infrastructure must be encrypted and your system should use logical and process isolation to ensure sensitive data isn't erroneously sent to the wrong destination.



Probable costs of an automated solution

The alternative to building a DIY data pipeline is to purchase one. The cost of subscription for an automated solution may be a flat yearly rate or based on consumption. There are many different kinds of pricing available, such as a schedule based on monthly active rows (MAR).

The purpose of automation is fundamentally to trade money for a much larger equivalent sum in labor and time. A good solution may allow you to begin syncing data into the warehouse in a matter of minutes or hours rather than weeks or months. More importantly, it should allow you to do considerably more with less in terms of money and talent. A good off-the-shelf solution can easily be the equivalent of conjuring up a data engineer or two out of thin air.

Chapter 5: How to build a modern data stack

We previously defined a data stack as the suite of tools and processes used to extract, load, transform and analyze data. The modern data stack leverages advancements in cloud-based technologies, third-party tools and automation. As previously described, the essential elements of a modern data stack include:

1. Data integration tool
2. Data warehouse
3. Business intelligence platform

These elements may include support for transformations, data activation and machine learning, as well. Together, the cloud-based elements of a modern data stack make data-driven decisions radically simpler to implement than with traditional on-premise or legacy technologies.

Key business considerations

For each element of your data stack, consider the following factors:

1. **Pricing and costs** – Make sure the pricing and cost schedules for each tool make sense for your organization. Be mindful of the total cost of ownership as well as the opportunity costs of alternatives, especially DIY solutions.
2. **Fit to team's skills and future plans** – Your team should be capable of using the tools and technologies in question. For instance, your analysts might be best situated to perform transformations using SQL rather than a scripting language.
3. **Vendor lock-in and future-proofing** – Can you continue integrating data even if a vendor exits the industry or changes its terms of service?

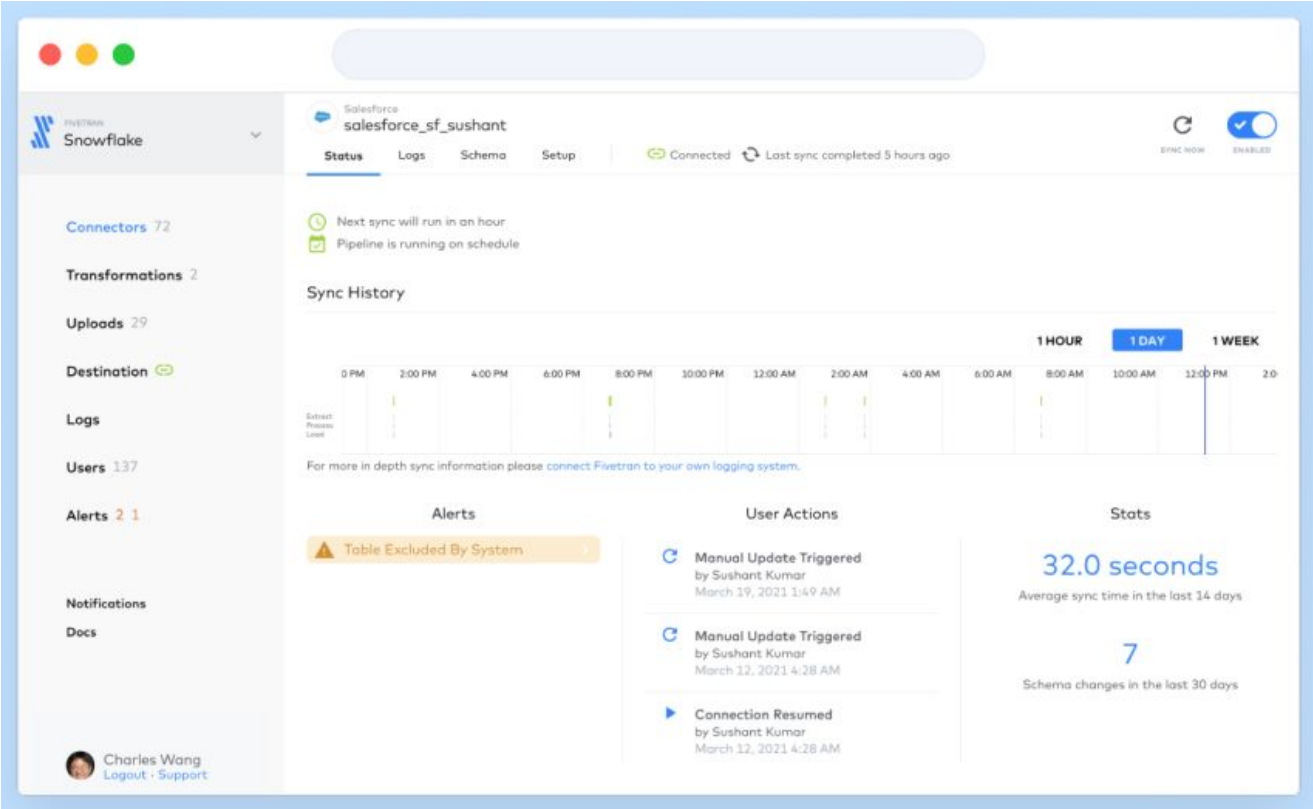
These are organizational considerations that have more to do with how you plan to grow and sustain your organization in the future than the technical characteristics of each tool.

Choosing the right data integration tool

There are many data integration tools in the market, and their technical approaches and feature sets vary significantly. Here are the foremost factors to consider when choosing a data integration tool:

1. **Data connector quality** – Take these factors into account when evaluating connector quality:
    a. **Open-source vs. proprietary** – There are 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.
    b. **Standardized schemas and normalization** – Data from API feeds is not usually normalized. Normalization fosters data integrity, while standardization enables outsourcing and automation by allowing a provider to support the same solution for a wide range of customers.
    c. **Incremental vs. full updates** – Incremental updates using logs or other forms of change detection allow for more frequent updates that do not interfere with business operations.
2. **Support for sources and destinations** – Does the tool support your sources and destinations? Does the provider offer a way for customers to suggest new sources and destinations? Do they routinely add new ones?
3. **Configuration vs. zero-touch** – Zero-touch, fully managed tools are extremely accessible, with connectors that are standardized, stress-tested and maintenance-free. By contrast, configurable tools require expensive allocations of engineering time.
4. **Automation** –Integration tools should remove as much manual intervention and effort as possible. Consider whether a tool offers features like automated schema migration, automatic adjustment to API changes and continuous sync scheduling. Machines are generally cheaper than humans, and the purpose of automation is to exploit this advantage.
5. **Transforming within the data warehouse** – With an ELT architecture, analysts can perform SQL-based transformations in an elastic, cloud-based warehouse. SQL-based transformations also offer the possibility of jumpstarting analytics using off-the-shelf SQL-based data models.
6. **Recovery from failure** – You don't want to ever permanently lose data. Find out whether your prospective tools are idempotent and perform net-additive integration.

7. **Security and compliance** – These are key areas, both in terms of data protection and public perception. Specifically, learn how prospective tools deal with:
    a. Regulatory compliance
    b. Limited data retention
    c. Role-based access
    d. Column blocking and hashing

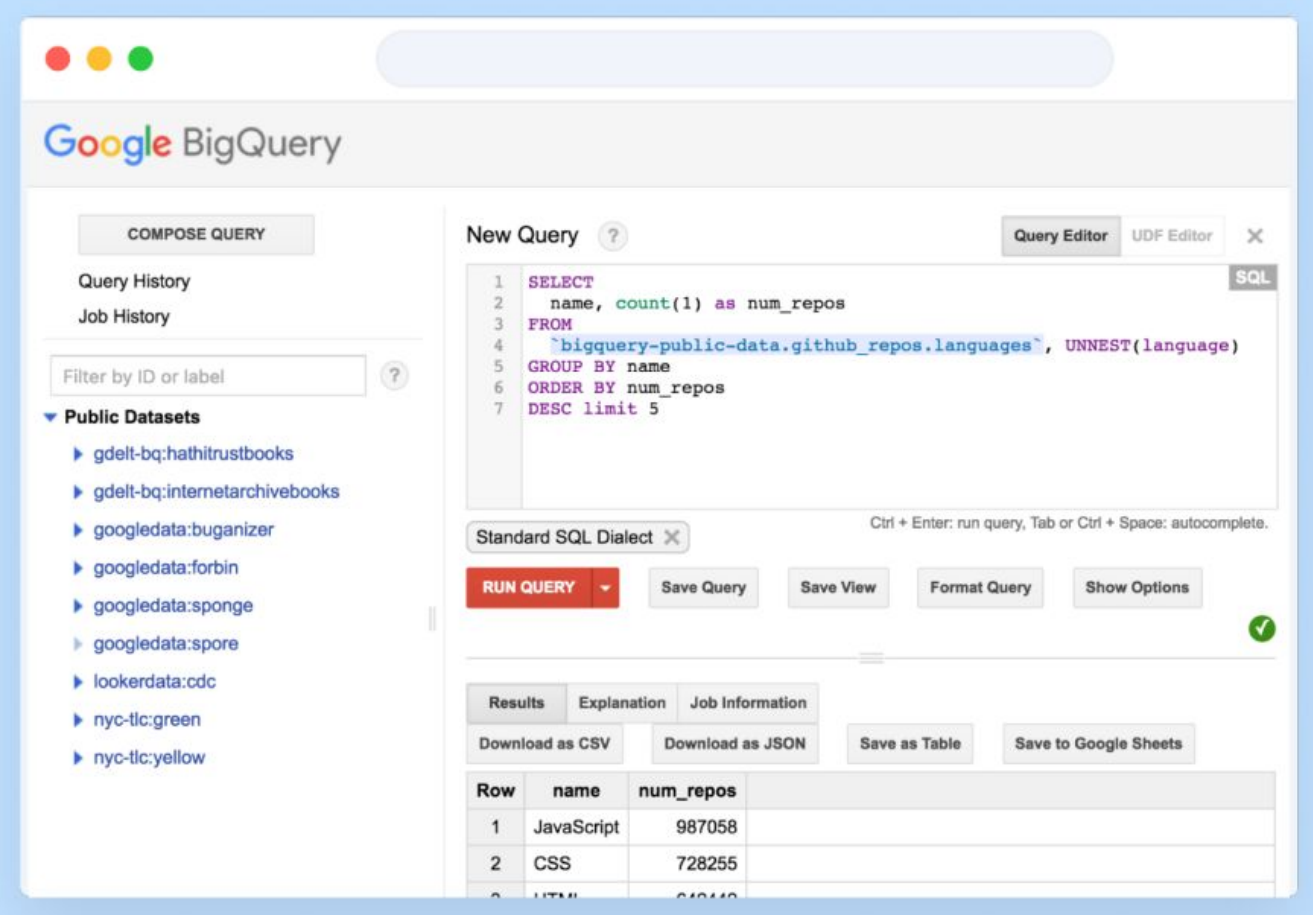


Fivetran offers a zero-touch approach to data integration into common data warehouses like Snowflake.

Choosing the right data warehouse

Your data warehouse will be the repository of record for your organization's structured data. Different data warehouses offer different features and tradeoffs. Here are the nine criteria you should focus on:

1. **Centralized vs. decentralized data storage** – Does the data warehouse store all of its data on one machine, or is it distributed across multiple machines, trading redundancy for performance?
2. **Elasticity** – Can the data warehouse scale compute and storage resources up and down quickly? Are compute and storage independent from each other or coupled together?
3. **Concurrency** – How well does the data warehouse accommodate multiple simultaneous queries?
4. **Load and query performance** – How quickly can you complete typical loads and queries?
5. **Data governance and metadata management** – How does the data warehouse handle permissions and regulatory compliance?
6. **SQL dialect** – Which dialect of SQL does the data warehouse use? Does it support the kinds of queries you want to make? Will your analysts have to adjust the syntax they currently use?
7. **Backup and recovery support** – If your data warehouse somehow gets corrupted or breaks, can you easily revert to a previous state?
8. **Resilience and availability** – What about preventing database failures in the first place?
9. **Security** – Does the data warehouse follow current security best practices?



Data warehouses have similar interfaces to those of operational databases.

As a repository for your data, your choice of data warehouse may very well be the most financially impactful part of your data stack. Upgrades and changes to your data warehouse involve data migrations and are very costly.

Choosing the right business intelligence tool

Business intelligence tools enable you to easily build reports and dashboards, but different tools have different strengths and weaknesses. Here are the key factors to consider:

1. **Seamless integration with cloud data warehouses** – Is it easy to connect this BI tool to your cloud data warehouse of choice?
2. **Ease of use and drag-and-drop interfaces** – Ease of use is especially important to popularizing data-driven decisions across your organization.

3. **Automated reporting and notifications** – Writing reports by hand can become tedious for data scientists and analysts. Does the BI tool allow you to schedule reports to publish automatically? What about alerting users when the data changes?
4. **Ability to conduct ad hoc calculations and reports by ingesting and exporting data files** – Your analysts and data scientists might sometimes want to explore data without the overhead of having to go through a data warehouse first.
5. **Speed, performance and responsiveness** – Basic quality-of-life considerations are important, like dashboards and visualizations loading in a timely manner.
6. **Modeling layer with version control and development mode** – Does the BI tool allow your analysts to work collaboratively by sharing data models and code?
7. **Extensive library of visualizations** – Pie charts, column charts, trendlines and other basic visualizations can only take you so far. Does the BI tool feature more specialized visualizations like heat maps or radar charts? Does it allow you to build your own custom visualizations?

A business intelligence tool might support bar, column and pie charts, trend lines, heat maps and many other types of visualizations

Chapter 6: Six steps to getting started with data integration

There are six practical steps to any data integration journey:

1. Rule out barriers to a modern data stack
2. Migrate or start fresh
3. Evaluate the elements of your modern data stack
4. Calculate total cost of ownership and ROI
5. Establish success criteria
6. Set up proof of concept

Rule out barriers to a modern data stack

The modern data stack depends on outsourcing and automating your data operations, but there are legitimate reasons not to involve third-party or cloud-based providers.

The first and most obvious reason 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 tiny 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 nanoseconds of latency can make or break your operations, you might avoid third-party cloud infrastructure and build your own hardware.

A third reason is that your organization is in the business of producing its own specialized software products, and using or selling the data produced by their software. What if you are a streaming web service that produces terabytes of user data every day and also surfaces recommendations for users? Even so, your organization may still outsource data operations for external data sources.

Otherwise, if your organization is of 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 hassle because of the volume and intrinsic complexity of data. Whether your company decides to migrate or simply start a new instance from scratch depends heavily on how important historical data is to you.

It may be costly to end existing contracts for products or services. Beyond money, familiarity with and preference for certain tools and technologies can be an important consideration.

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

Evaluate the elements of your modern data stack

You will need a data integration tool, data warehouse, business intelligence platform and transformation layer. Refer to the previous chapter for the exact criteria you should use to evaluate your choices and make sure the technologies are compatible with each other.

Calculate total cost of ownership and ROI

The modern data stack promises substantial savings of time, talent and money. Compare your existing data integration workflow with the alternatives.

Calculate the cost of your current data pipeline. The main factor is likely to be the amount of engineering time your data team spent building and maintaining the data pipeline. This may require a careful audit of tools you use for project management.

You'll need to consider the sticker price of the tools and technologies involved as well. Finally, you'll also need to consider any opportunity costs incurred by failures, stoppages and downtime. Include the costs of your data warehouse and BI tool as well.

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

An automated data integration solution can serve a number of goals. Base your success criteria on the following:

1. **Time, money and labor savings** – A modern data stack should dramatically reduce your data engineering costs by eliminating the need to build and maintain data connectors. Labor savings may amount to hundreds of hours of engineering time per week, with the corresponding monetary figures. You can **use our calculator** to get a high-level estimate.
2. **Expanded capabilities** – A modern data stack (MDS) should expand the capabilities of your data team by making more data sources available without additional labor.
3. **Successful execution of new data projects, such as customer attribution models** – More time and data sources allow your team to build new data models, including those that track the same entities across multiple data sources.
4. **Reduced turnaround time for reports** – A modern data stack should dramatically shorten the turnaround time for reports, ensuring that key decision-makers stay up to date.
5. **Reduced data infrastructure downtime** – A modern data stack should dramatically improve reliability and virtually eliminate your maintenance burden.
6. **Greater business intelligence usage** – By combining automated data integration with a modern, intuitive BI tool, a modern data stack should promote data access, literacy and usage across your organization.
7. **New available and actionable metrics** – With additional data sources and an easy-to-use BI tool, a modern data stack should enable new metrics and KPIs for decision-making.

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 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.

While you may have ruled out technical barriers, there can be institutional barriers to adopting a modern data stack. Your data team could lack funding, headcount or expertise. Data engineers might be protective of the systems they have built. Leaders might not recognize the power offered by the ability to rapidly scale data integration. It is important to earn buy-in from someone with the authority to purchase the necessary tools and technologies, and to carefully cultivate a modern data mindset from the very start of your journey.

A carefully constructed minimum viable product (MVP) demonstration that proves the worthiness of the modern data stack on a single data source, report or test case can accomplish exactly that.

Chapter 7: How to continue modernizing your analytics

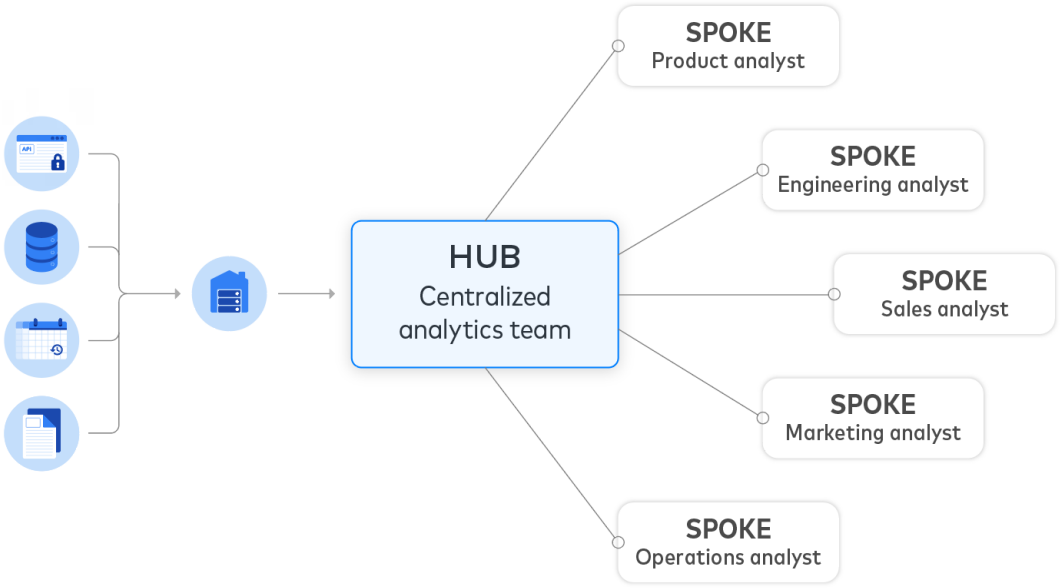
A modern data stack is just the first step to building a mature data operation. With the right tools in place, your data team will be able to extract, load, model and transform your data. However, it will take progressive organizational changes to expand and continue making good use of your new capabilities. These changes include:

1. Scale your analytics organization
2. Establish data governance standards
3. Use product thinking
4. Promote data literacy
5. Build a robust data architecture
6. Hire data scientists

Scale your analytics organization

One of your first considerations after building a modern data stack is scaling your data team. The bread and butter of analytics teams are analysts, whose expertise enables them to build data models, dashboards and reports to help your organization make decisions.

As your data needs grow and become more complex, you will need to expand your roster of analysts. There are arguments in favor of both centralized and decentralized data teams, but a good compromise is both, in the guise of a **hub-and-spoke model.**



Your "hub" is a centralized team that owns the overall data integration process and creates and maintains the less specialized data models, reports and visualizations used by company leadership and individual contributors alike. The "spokes" are small, functionally-aligned teams of analysts who are embedded with specific departments and have relevant domain expertise in areas such as sales, finance and so forth. The hub and spokes alike should report to company leadership. It is worth reserving a seat in the C-Suite for this – a chief analytics officer, chief data officer or equivalent.

Establish data governance standards

The other major process consideration is **data governance**. Ownership of data assets also means carefully documenting and cataloging all data assets. This becomes more important the more data you ingest and the more products or lines of business you develop. An ungoverned "data swamp" can turn your data unusable. It can also make compliance with regulatory standards (and basic ethical considerations) difficult or impossible.

To avert this problem, consider a cloud data catalog tool. It will help you take the following actions:

- Document all models, tables and fields. This may be impractical if you have many data sources; an alternative is to carefully build a dimensional schema, which is a simplified data model that encompasses all major operations.
- Determine what metrics you need and where they come from.
- Make note of how frequently you need to refresh the data.
- Plan to address any data integrity issues.
- Identify the true data owners for the various models within the organization.
- Assign ownership and create incentives to keep the system healthy.

The best time to make this effort is as you start fully implementing your modern data stack, as you will need to take inventory of all data assets anyway.

Get data governance under control early in order to build trust. Without a clear provenance for every data model, it will be difficult for the end users of your data to make sense of how metrics are determined and resolve conflicting narratives.

Use product thinking

As your data team becomes more heavily involved in data visualization and decision support, you will need to make a concerted effort to bring product thinking to your analytics efforts. You will need a new role called a data product manager to lead the creation of data assets.

In a nutshell, product thinking means understanding your users (i.e. individual contributors and leadership in your organization) and rapidly, iteratively producing and refining products in response to changing conditions. In a little more detail, the product process for data assets involves:

Identify

- Understanding users
- Gathering requirements

Design

- Defining scope
- Managing expectations

Develop

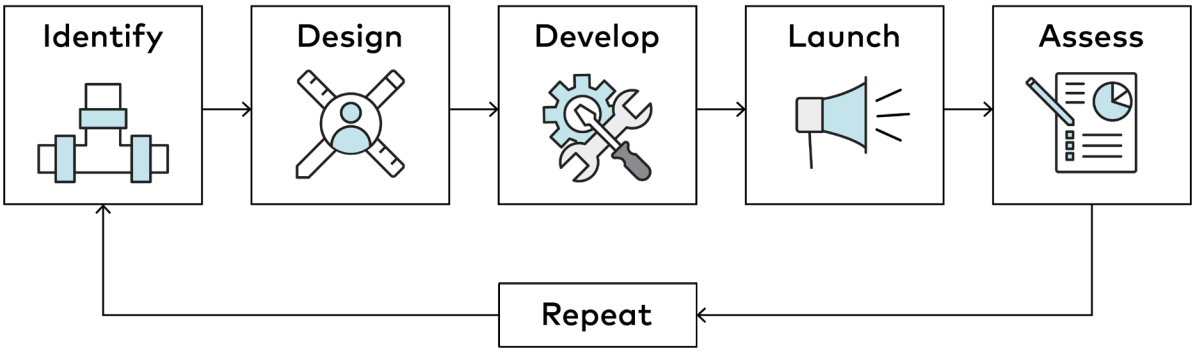
- Rapid prototyping
- Productionizing

Launch

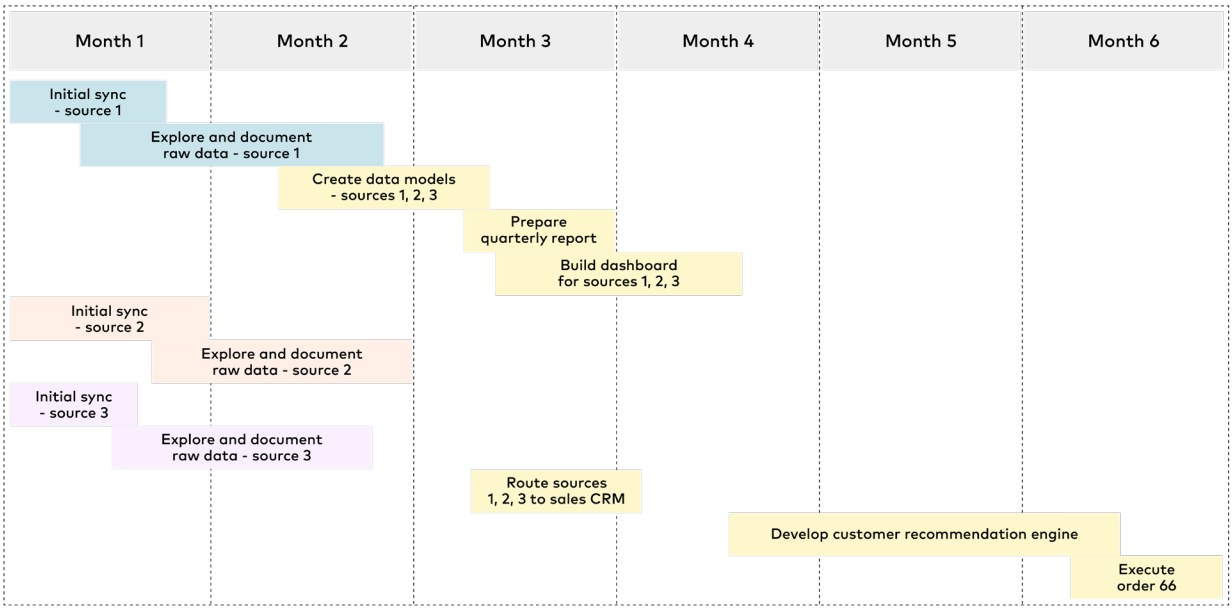
- Marketing and rolling out the product
- Training users via office hours and internal communications
- Driving adoption, including through self-service whenever possible

Assess

- Evaluating against expectations and KPIs



You will need to build a roadmap regarding how data assets will improve decision-making at your organization. List the milestones and end goals, as well as the information and insights necessary to achieve them. Your roadmap is something you can bring to your leadership with the possibility of being incorporated into your organization's strategy.



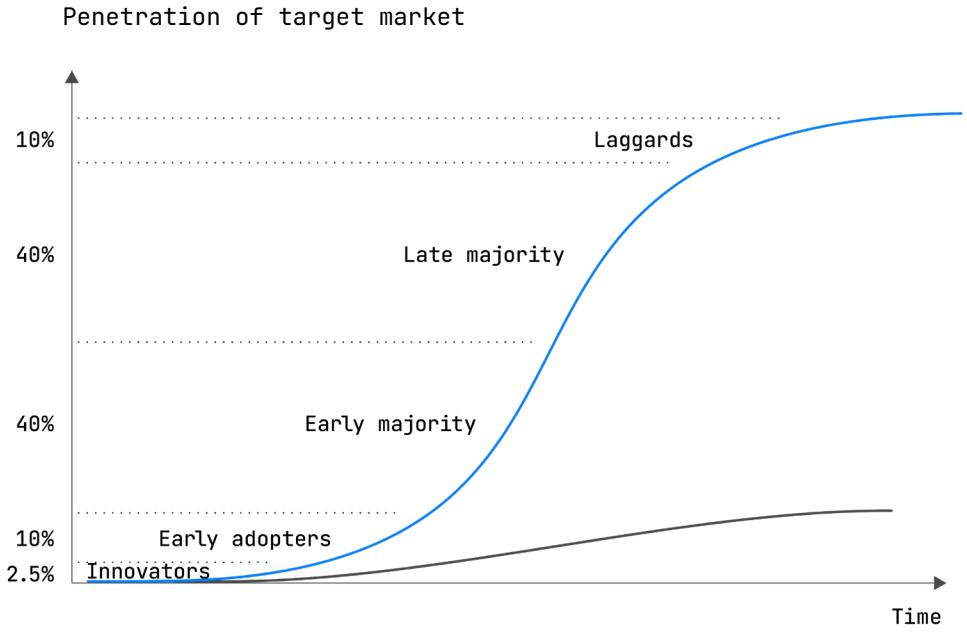
There are some specific considerations when it comes to messaging a data-driven agenda. Those who lack expertise in navigating the appropriate tools and metrics to analyze data can be skeptical. They may even have dystopian preconceptions of how data can be weaponized for unethical or otherwise dubious purposes.

It is important to recognize these dangers in advance, and to communicate clearly about the benefits of becoming data-driven in clear terms that appeal to your audience's priorities, not just your own. Avoid buzzwords and deep technical details in favor of emphasizing how desired, data-driven outcomes are achieved. Data-based storytelling is key, and real examples or proof points are invaluable – you will have to seamlessly combine strategy, visualizations and narratives. This will inspire people to become data literate and begin using data.

Promote data literacy

As your data team promotes product thinking, you will also need to spread data literacy across both leadership and individual contributors. Some people use the phrase "citizen analyst" to describe distributed decision-making. It isn't reasonable to demand analysts and other data professionals to interpret everything on behalf of decision-makers.

It can be useful to think of data literacy in terms of the **diffusion of innovation theory**. Imagine an S-shaped curve in which the leading 2.5 percent of people are innovators and trailblazers; another 13.5 percent are early adopters; another 34 percent are the early majority. The key challenge to getting started is to find an enthusiastic and technically-savvy person in a position of influence to evangelize to their team on your behalf. As teams become more data literate and more capable, your efforts should eventually snowball and gain momentum of their own.

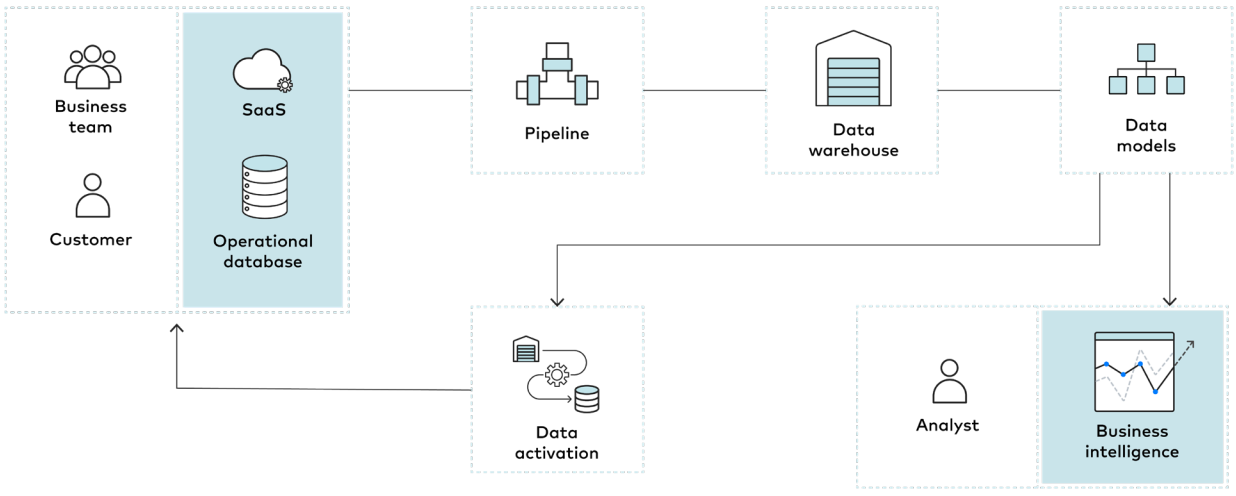


Going forward, making data literacy a key criterion for hiring will make it easier to improve the data capabilities of your teams. Remember, the baseline isn't writing SQL or building new models but a general ability to interpret graphs and tables and make decisions accordingly.

Build a robust data architecture

Data architecture refers to the full arrangement of tools and processes used for data integration. As your data operations mature, you will add new tools and technologies, use additional features of existing tools, make organizational changes and create new workflows. It will become ever more important to "close the loop" by activating analytics data and turning insights into business decisions and impactful actions. Specifically, you will need to consider:

- **Scaling to easily accept new data sources** – Make sure your data team and the tools they use are able to expand your data integration and analytics efforts to a wide variety of data sources. This means finding a scalable way to produce data models and ensuring your data team has domain expertise in a number of different business functions.
- **Automated reporting** – As your organization scales, you shouldn't count on analysts to produce reports or dashboards to order when many modern BI platforms can do so on a schedule.
- **Programmatic control over data integration** – As your users and data sources grow in number, you may want a way to programmatically control schedules, assign permissions and manage your data pipelines.
- **Activating data for use in production** – You will need a combination of off-the-shelf tools and data engineering expertise to bring analytics data into operational and production systems.



Hire data scientists

Congratulations! You have reached the top of the data hierarchy of needs we covered in Chapter 1. You are now ready to hire data scientists, and even begin to leverage AI and machine learning for exponential amounts of data analysis and game-changing insights.

Data scientists combine expertise in applied statistics and linear algebra with enough engineering chops to prototype machine learning models, bringing them into production with the help of data engineers.

Many organizations jump the gun and hire data scientists to do the work of analysts or data engineers before building a fully-fledged data hierarchy. This is a mistake. It is best to hire data scientists as your data infrastructure matures and after your organization has fulfilled more basic data needs.

Assign data scientists to design, build, test and tune machine learning models as your organization pursues predictive modeling and artificial intelligence.

By now you should have a full appreciation of the modern data stack, and the modern data mindset required to build it into your organization, surrounded by motivated teams and supportive leadership. Good luck on your journey, and thank you for reading!

Fivetran is the global leader in modern data integration. Our mission is to make access to data as simple and reliable as electricity. Built for the cloud, Fivetran enables data teams to effortlessly centralize and transform data from hundreds of SaaS and on-prem data sources into high-performance cloud destinations. Fast-moving startups to the world's largest companies use Fivetran to accelerate modern analytics and operational efficiency, fueling data-driven business growth. For more info, visit **Fivetran.com**.