



Cracking the JSON Code

A Guide to Managing and Analyzing JSON
with Snowflake and Sigma



[Introduction](#)

[The Rise of JSON](#)

[A New Way to Ingest, Store and Manage JSON](#)

- [Schema-on-Read Holds Back Timely Analysis](#)

[Snowflake Changes the Game](#)

- [Introducing VARIANT: A New Data Type](#)
- [Extracting Data Insights with SQL Extensions](#)
- [A Cloud Data Platform](#)

[Analyzing JSON Without Writing SQL](#)

- [Sigma Cracks the Code](#)
- [Visually Extract and Explore JSON without SQL in Sigma](#)
- [Write Back JSON Views to Snowflake for Future Analysis](#)

[Code Cracked](#)

A New Era for Data

The era of Big Data—and big data analytics—is upon us. By 2025 the global datasphere will grow to an estimated [175 zettabytes](#). For context, from the dawn of the internet to 2016, the web created a single [zettabyte of data](#).

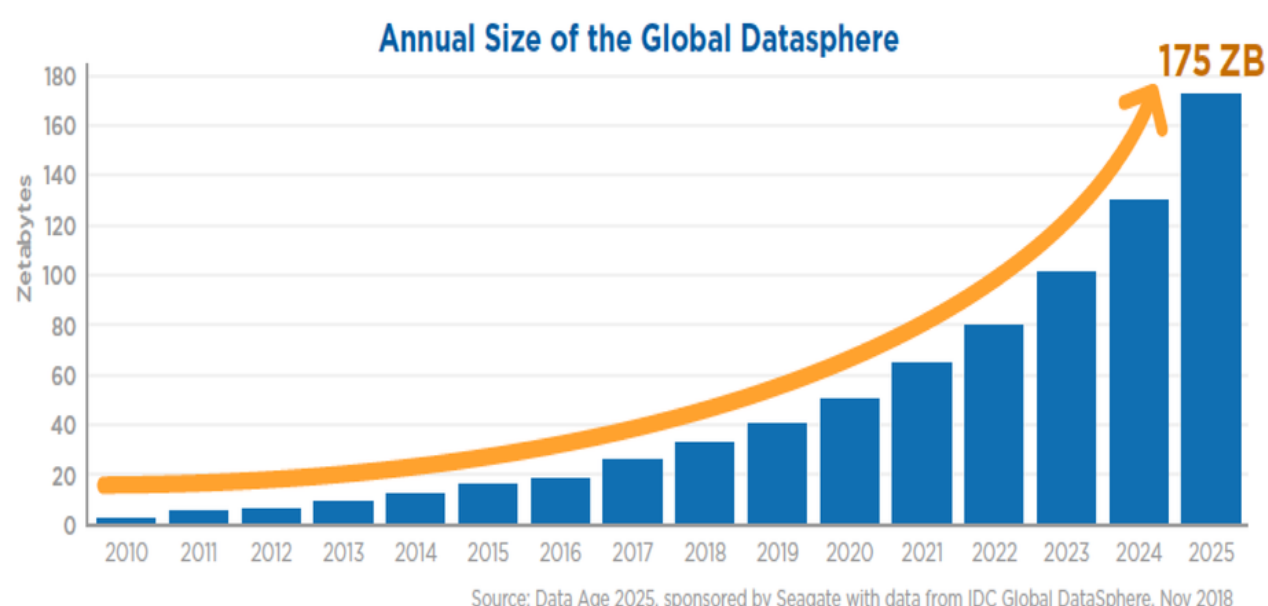
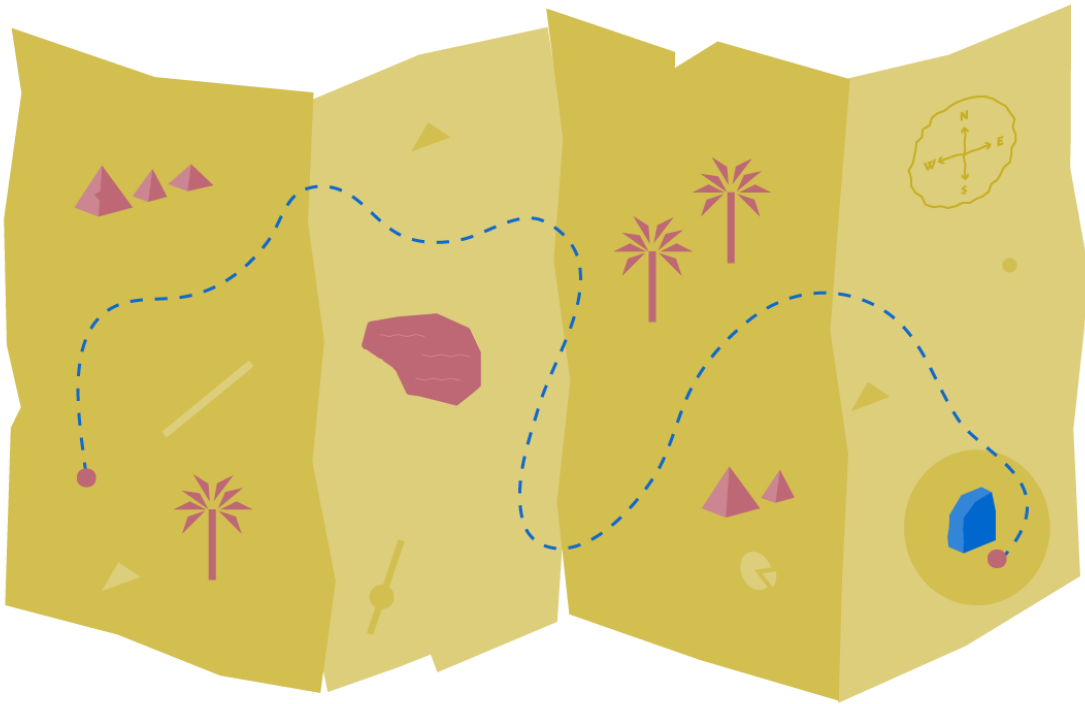


Image Source: [Forbes](#)

Of course, internet traffic is only a single slice of the data pie created and stored around the world — which includes all personal and business data. Today, the world sits somewhere between 40–50 zettabytes of total data. Which begs the question: What do we do with all this data? And what good will come from the constant collection of data across the web, personal devices, Internet of Things (IoT), and more? If you said, “analyze it for insights,” you’re on the right track. Unfortunately, [up to 73%](#) of all enterprise data goes unused. As more organizations aim to become data-driven, it’s imperative to make better use of the data they collect.

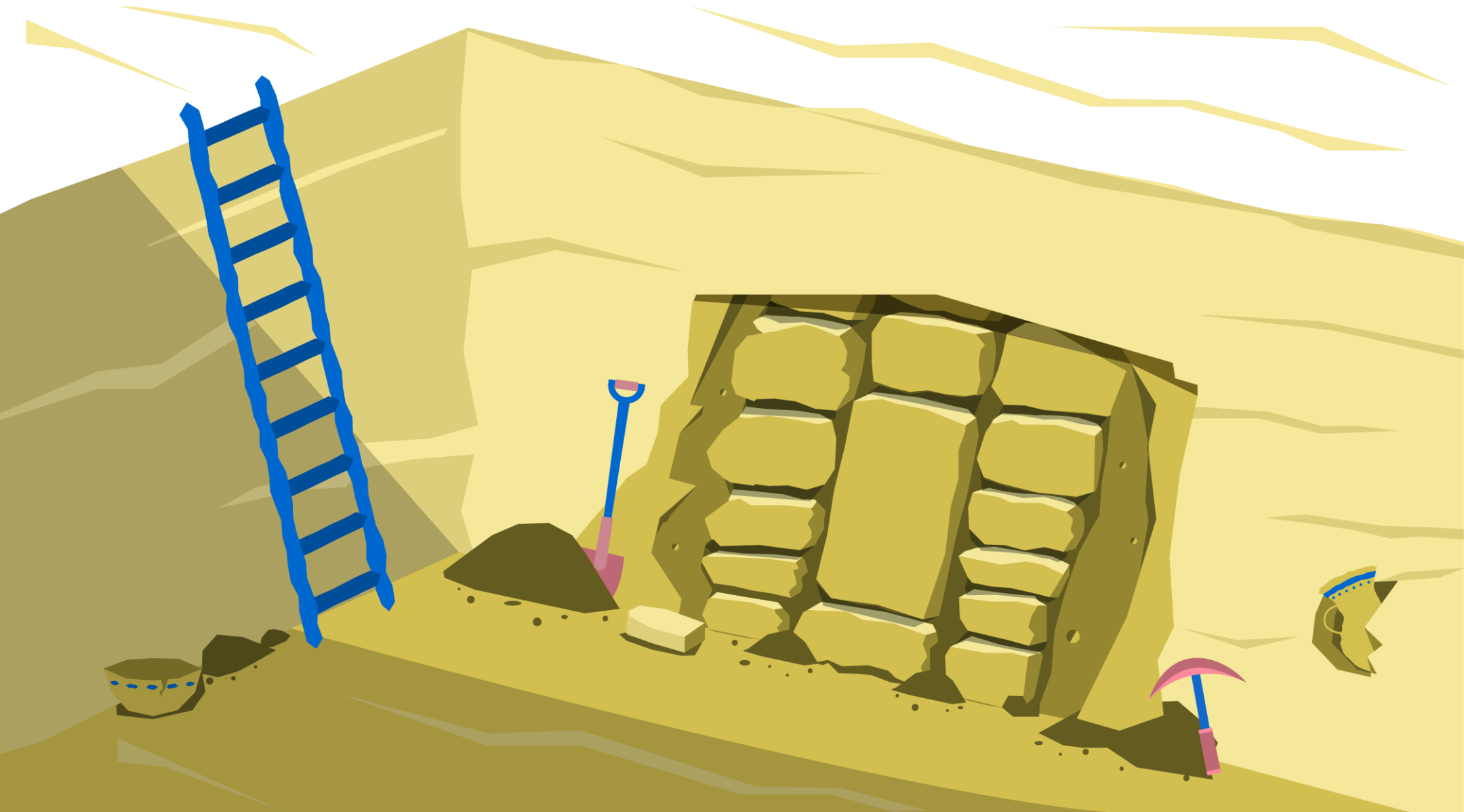
A good place to start is JSON. Since it arrived on the scene in 2001, JSON has become the preferred data interchange format for mobile devices, web applications, online services, and sensors. This includes some of today’s most popular websites like Facebook and Google — and the fast-growing market of wearables and IoT devices. These services and devices produce an unprecedented amount of data in our digital economy. Unstructured and semi-structured data (like JSON) now make up an [estimated 80%](#) of data collected by enterprises. And that number is only expected to grow in the coming years.



All this data is a potential treasure trove for companies that can harness it effectively. But combing through JSON in real time to find patterns, emerging trends, and insights has historically taken significant time and resources.

JSON analysis presents challenges for companies that want to make sense of this vast repository of information. JSON can't be stored, managed, and analyzed as quickly as [structured data formats](#) — which has data experts leaving valuable insights on the table. And if you're a marketer, salesperson, or product leader, you can't extract and analyze JSON without significant hand-holding from your colleagues on the data team. Even for those with a background in data science, the process of parsing JSON is cumbersome.

With Snowflake+Sigma, it's possible for JSON analysis to extend beyond data and IT teams. Domain experts can now parse JSON, join it with other structured data, and generate insights to drive better decisions — and get a leg up on the competition.



The Rise of JSON

For those less familiar with JSON, or [JavaScript Object Notation](#), it is a schema-less, text-based representation of structured data that is based on key-value pairs and ordered lists. JSON has become extremely popular over the last decade. Today, any time web applications communicate, it's usually through JSON. Internet giants like Google, Facebook, and Twitter all rely on JSON to expose data through APIs.

While there are other semi-structured data formats (such as XML and AVRO), none are more popular than JSON. It's become the de-facto way that programs interchange data in the modern era. So how – and why – did this happen?

The first JSON message was sent in 2001 from a server to a Douglas Crockford's laptop in Chip Morningstar's garage, which was the birthplace of their startup called State Software.

2001

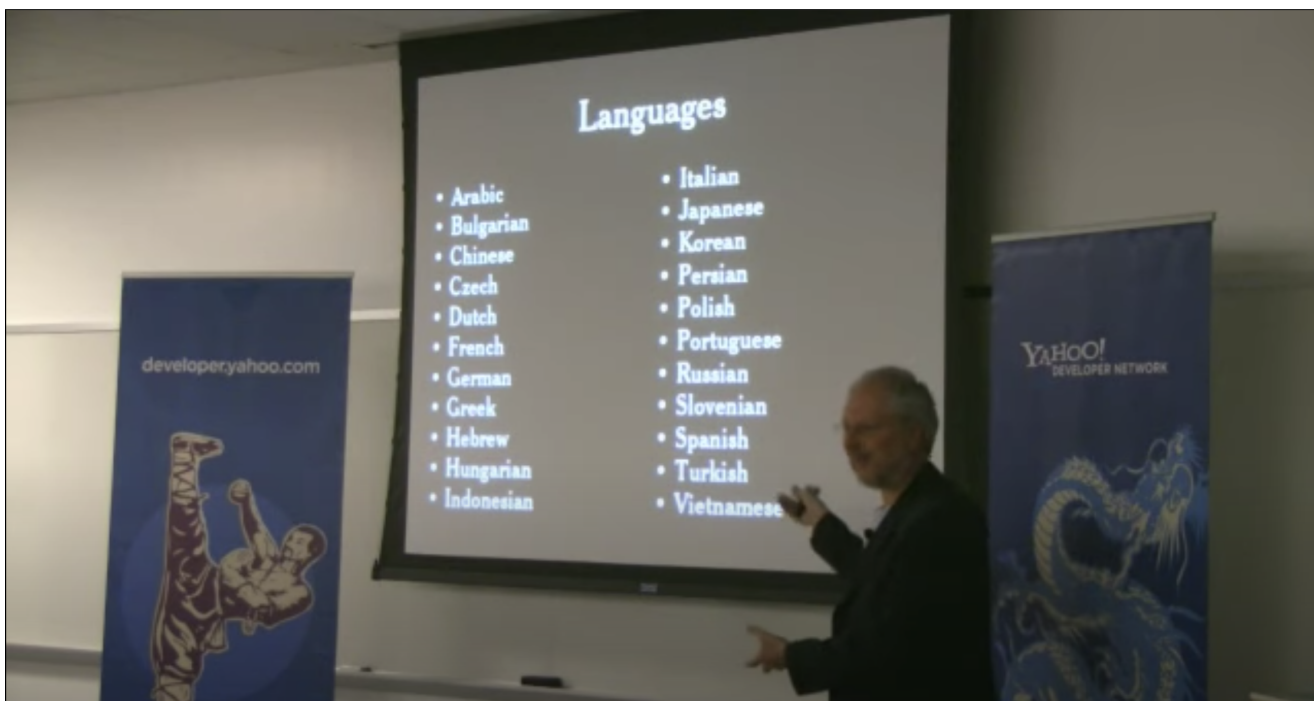
The first JSON message was sent from a server to Douglas Crockford's laptop in Chip Morningstar's garage. [\(source\)](#)

```
<html><head><script>
document.domain = 'fudco';
parent.session.receive(
  { to: "session", do: "test",
    text: "Hello world" }
)
</script></head></html>
```

The first JSON message sent in 2001

Crockford and Morningstar realized they could sidestep an HTML frame and send themselves form fill data. They found that this new method of data interchange was an efficient way to communicate data between servers, and even build a database. It was simple and represented an intersection of all modern programming languages.

They decided to call this new interchange format JSON, and registered json.org to get the word out. Soon other programmers started implementing JSON code across the web in various applications and sharing their work. Here's a video where Crockford recounts the early days of JSON.



[Video](#) : Yahoo! JavaScript architect Douglas Crockford tells the story of how JSON was discovered and why it became a major standard for describing data.

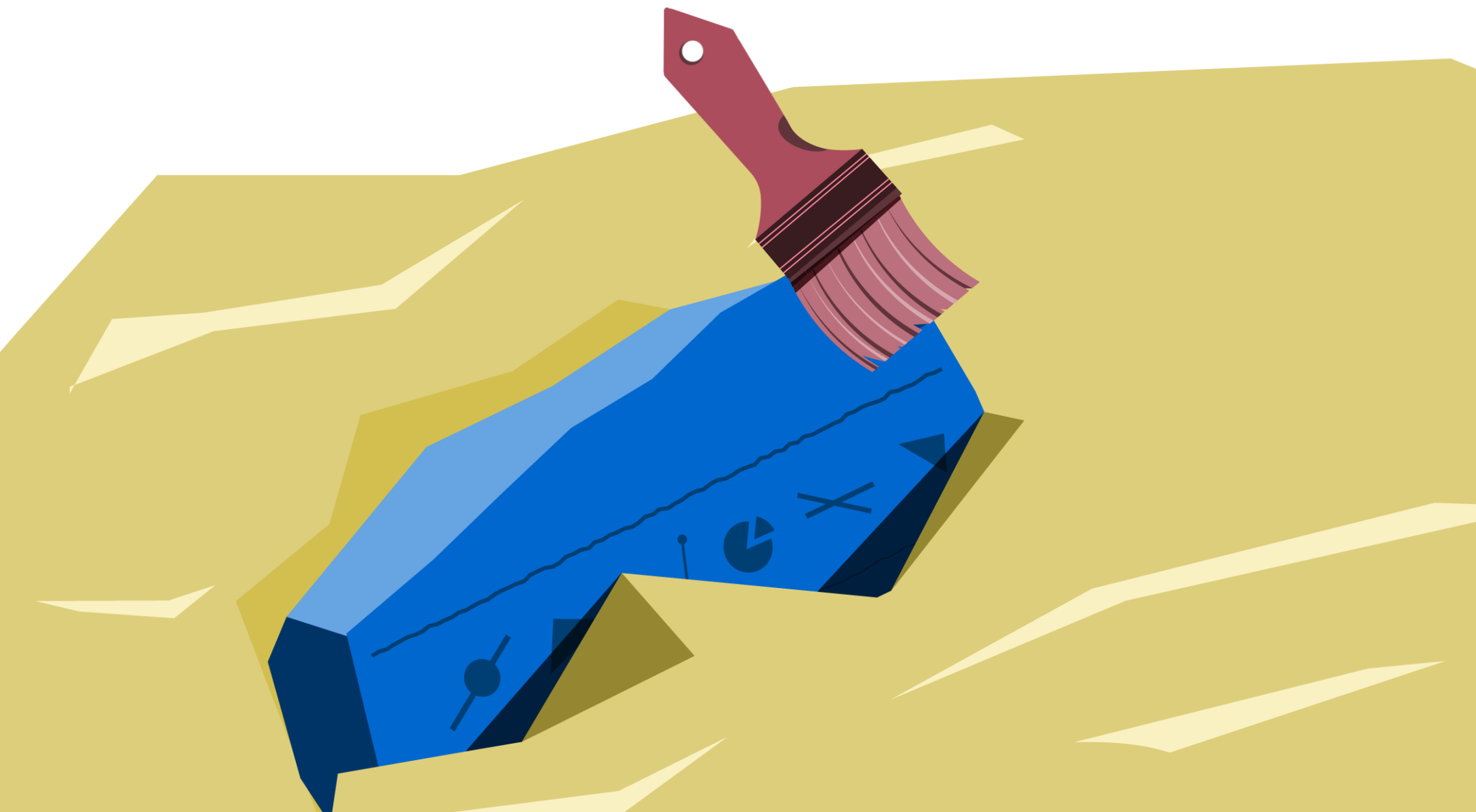
While exciting, JSON remained a niche language until 2005 when [AJAX](#) took off, allowing web pages to host entire applications without having to replace the page. Web developers began implementing JSON instead of XML – the dominant form of semi-structured data interchange at the time – because they found XML to be verbose and redundant, whereas JSON was much simpler and efficient. JSON is also human and machine-readable, making it easier to understand because its syntax is minimal, and the structure is predictable.

JSON	XML
Text based format (Not a Language)	Markup Language
Free to define anything	Has some rules
Smaller Size	Big in Size due to markups
JSON is similar to Java script Objects Literals. Browser read faster.	Browser need parsers to handle XML. Slow processing.
No support on namespaces and comments	Both are supported.

The rest, as they say, is history. JSON began to see widespread adoption in web applications everywhere and has since become the preferred method to interchange information between programs and devices connected to the internet (displacing XML). Below you can see a visual representation of developer interest in JSON compared to XML from Stack Overflow, a popular open community for developers.



Developer interest in JSON compared to XML 2009-2020. Source: [Stack Overflow](#)



A New Way to Ingest, Store and Manage JSON

JSON comes with its challenges. JSON contains readable tags and an implicit organizational structure, but it's a far cry from the rows and columns and rules of structured data. The structure is not fixed, and the files can contain an arbitrary depth of nesting. Flattening this data on its way into the data warehouse can result in lower resolution data and the loss of any detail that doesn't conform to a predefined relational schema. Conversely, having semi-structured data like JSON in the warehouse creates a semantics challenge for anyone trying to curate the data for downstream consumers.

Traditionally, ingesting, storing, and managing JSON with legacy data warehouses has been complicated and expensive. The data usually have an irregular and partial structure, and some sources have an implicit structure of data, which makes it challenging to interpret the relationship between data. Meanwhile, schema and data are usually tightly coupled. The same query may update both schema and data, with the schema getting updated frequently. This makes the distinction between schema and data uncertain or unclear, which complicates designing the data structure.

Schema-on-Read Holds Back Timely Analysis

In the past, your data team has likely turned to NoSQL platforms or Hadoop to store semi-structured JSON. If you've ever loaded JSON into a NoSQL database, you're probably familiar with the term **"schema-on-read."** With schema-on-read, you can delay data modeling and schema design until after the data gets loaded. This process speeds up the migration of data into the repository by eliminating the need to wait for a data modeler to design the tables.

Unfortunately, even with schema-on-read, more work has to be done before you can extract the data into a logical schema to be queried with standard SQL using an analytics tool. First, you would need to parse it with a platform like MapR and load it into columns within a relational database, which makes it possible for analytics and BI tools to run queries against the data. Writing code to extract data and determine the right schema slows down an organization's ability to access and analyze JSON quickly.

Snowflake Changes the Game

[Snowflake](#)'s patented technology natively loads and optimizes both structured and semi-structured data such as JSON, and makes it available via SQL without sacrificing performance or flexibility. By loading semi-structured JSON data directly into a relational table, it skips the entire schema-on-read process. Once data lives in a relational table, you can query and join it to other structured data using a standard SQL statement without worrying about changes to the data schema. Snowflake monitors the schema changes automatically, eliminating the need to rely on ETL or parsing algorithms.

Introducing VARIANT: A New Data Type

Snowflake solved this problem by inventing a new data type it calls VARIANT. [VARIANT](#) allows semi-structured data to be loaded into a relational table column as is. Snowflake automatically discovers data attributes and structure when it's loaded, and then uses that information to optimize storage — paying close attention to repeated attributes across records and storing them separately. This loading process compresses data more effectively and provides faster access to data, much in the same way a columnar database works. The VARIANT data type is not affected by changes over time, such as new attributes, nesting, or arrays. Snowflake's approach eliminates the need to add Hadoop or a NoSQL platform to your data warehouse architecture.

Extracting Data Insights with SQL Extensions

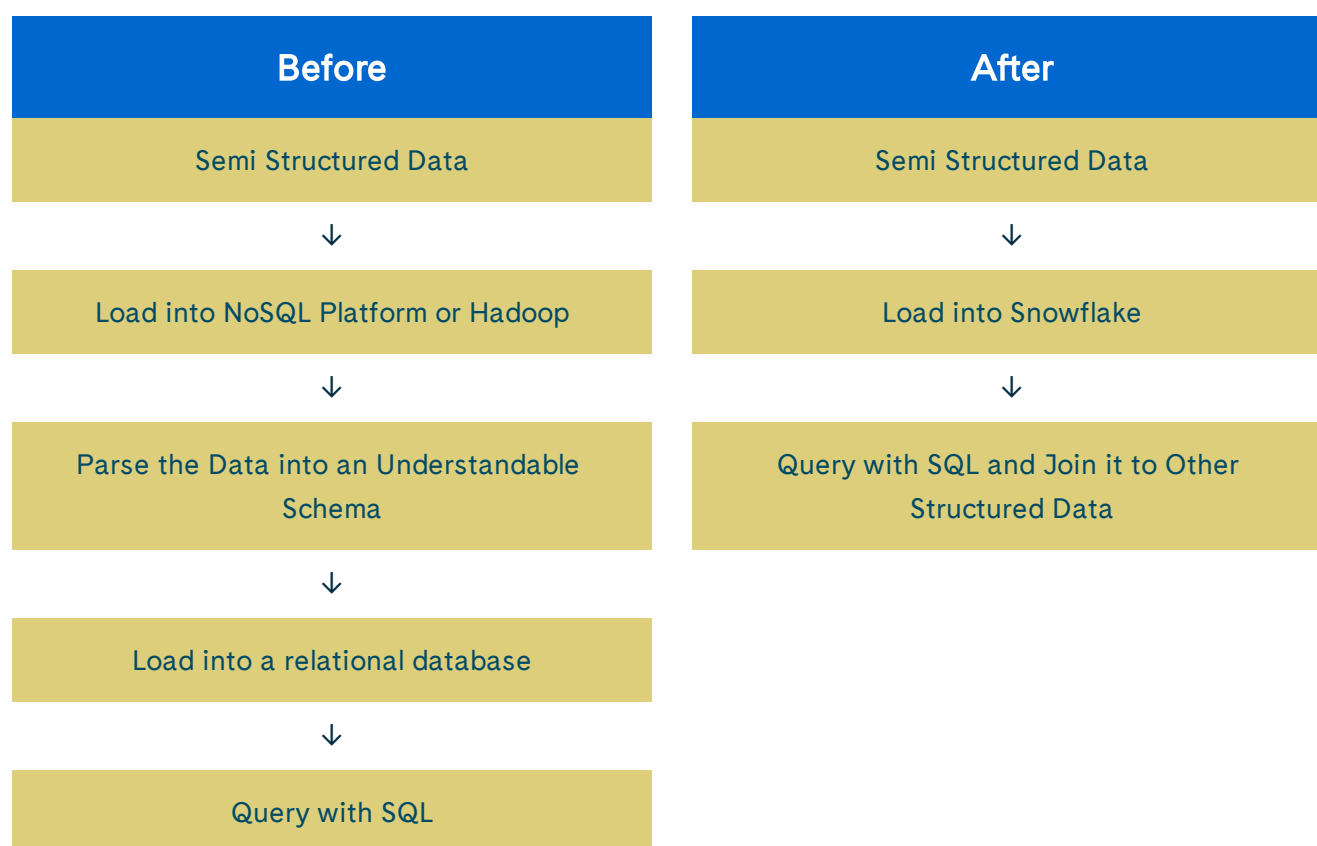
Once data is stored in Snowflake, it creates SQL extensions to reference the internal data schema. The self-describing SQL queries components and joins data to columns in tables similarly to parsing in a standard relational format — without the coding, ETL, or additional parsing usually required to prep the data.

Because Snowflake also collects, calculates, and stores statistics on sub-columns in its metadata repository, it can optimize data access with its advanced query optimizer. This allows Snowflake to minimize the amount of data required for access and speeds up the return of data in the process.



A Cloud Data Platform

Snowflake provides natively-optimized support for processing semi-structured JSON data in the cloud. Legacy solutions offered by traditional vendors cannot match the performance of Snowflake’s semi-structured data management. Even those with [cloud-washed](#) features to store and access JSON rely on legacy code and existing data types, requiring additional tuning by database admins to increase performance.



Snowflake eliminates the need to load JSON into NoSQL platforms or Hadoop and parse JSON before it gets to the RDBMS.

Snowflake helps you quickly and easily load semi-structured data into a modern data platform for immediate analysis.

For more information on Snowflake’s semi-structured data storage capabilities, including a step-by-step tutorial on loading semi-structured data, check out this [free eBook](#).



Analyzing JSON without Writing SQL

Even when you've solved the challenges of loading JSON into Snowflake, extracting nested JSON rows and analyzing them for insights still requires a deep technical background – meaning it's usually off-limits for those outside the data team. For those versed in SQL, the process can prove to be time-consuming. JSON typically requires sufficient data exploration and SQL to prep datasets for analysis. This is a heavy lift for the data team and holds back the line of business units from jumping in and exploring the data themselves.

Sigma Cracks the Code

[Sigma](#) is a new, cloud-native BI and analytics tool built for modern cloud data platforms, like Snowflake. Its familiar spreadsheet interface eliminates the technical barrier posed by traditional analytics tools, opening up cloud data to everyone for more in-depth analysis. Sigma turns user actions into SQL ‘under the hood,’ empowering domain experts to ask deeper questions without authoring it themselves. Business teams can explore vetted data and build ad hoc visualizations, dashboards, and reports in minutes without writing SQL — or asking for help from the data team. This visual approach to SQL makes it possible to analyze all types of data, even semi-structured data like JSON.



Sigma turns user actions into SQL ‘under the hood’, empowering domain experts to ask deeper questions without authoring it themselves.

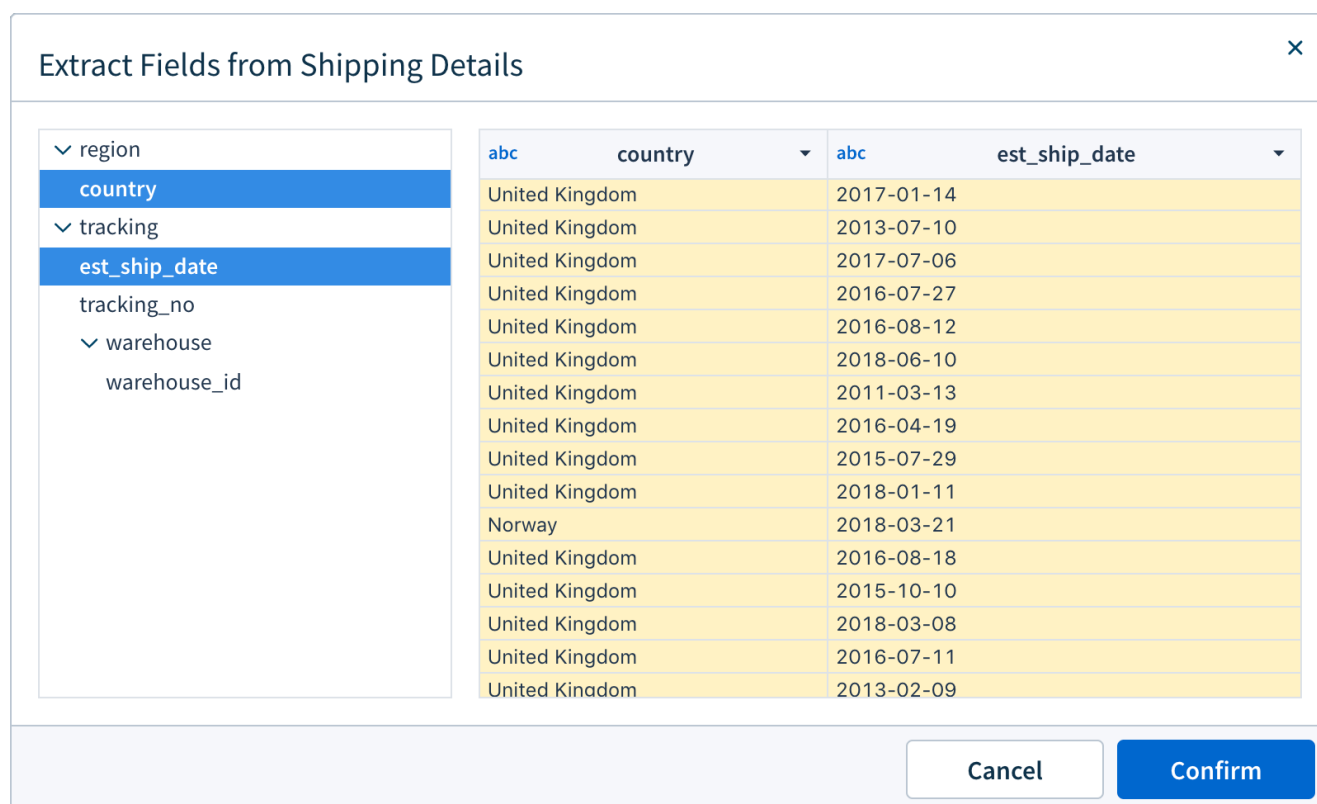
Sigma understands that analytics is all about improving business processes and outcomes — and that the best way to move that needle is to make it fast and easy for business experts to get their questions answered. The logic of these explorations and analyses is always expressible in SQL (or spreadsheet formulas), so users can interact with data however they're most comfortable. This facilitates better communication between data experts and business experts. Instead of requiring business experts to also become JSON and SQL experts or requiring data technicians to become experts in Salesforce, Netsuite, Marketo, and other business applications, Sigma provides a spreadsheet interface that connects directly to an enterprise’s cloud data warehouse.

This shift enables the data experts to spend more of their attention and creative energy on the technical work only they can perform, like systems architecture, analytical deep dives, performance tuning, governance, and data compliance.

[Sigma unlocks the benefits of semi-structured data](#) – expedience, transparency of data lineage, flexibility to add new fields, depth, breadth – without the classic trade-offs of cost, confusion, and technical training. By empowering the business experts to explore and comprehend raw data, Sigma helps businesses ask more of their data.

Visually Extract and Explore JSON without SQL in Sigma

Nested JSON often has multiple fields that may not be of interest during an ad hoc analysis. Using Sigma, users can easily extract semi-structured data fields and create relevant dataset views for exploration. When Sigma detects JSON or Variant column types, 'Extract Columns' becomes an option in the column menu. The extracted dataset view can then be analyzed in the Sigma Spreadsheet.



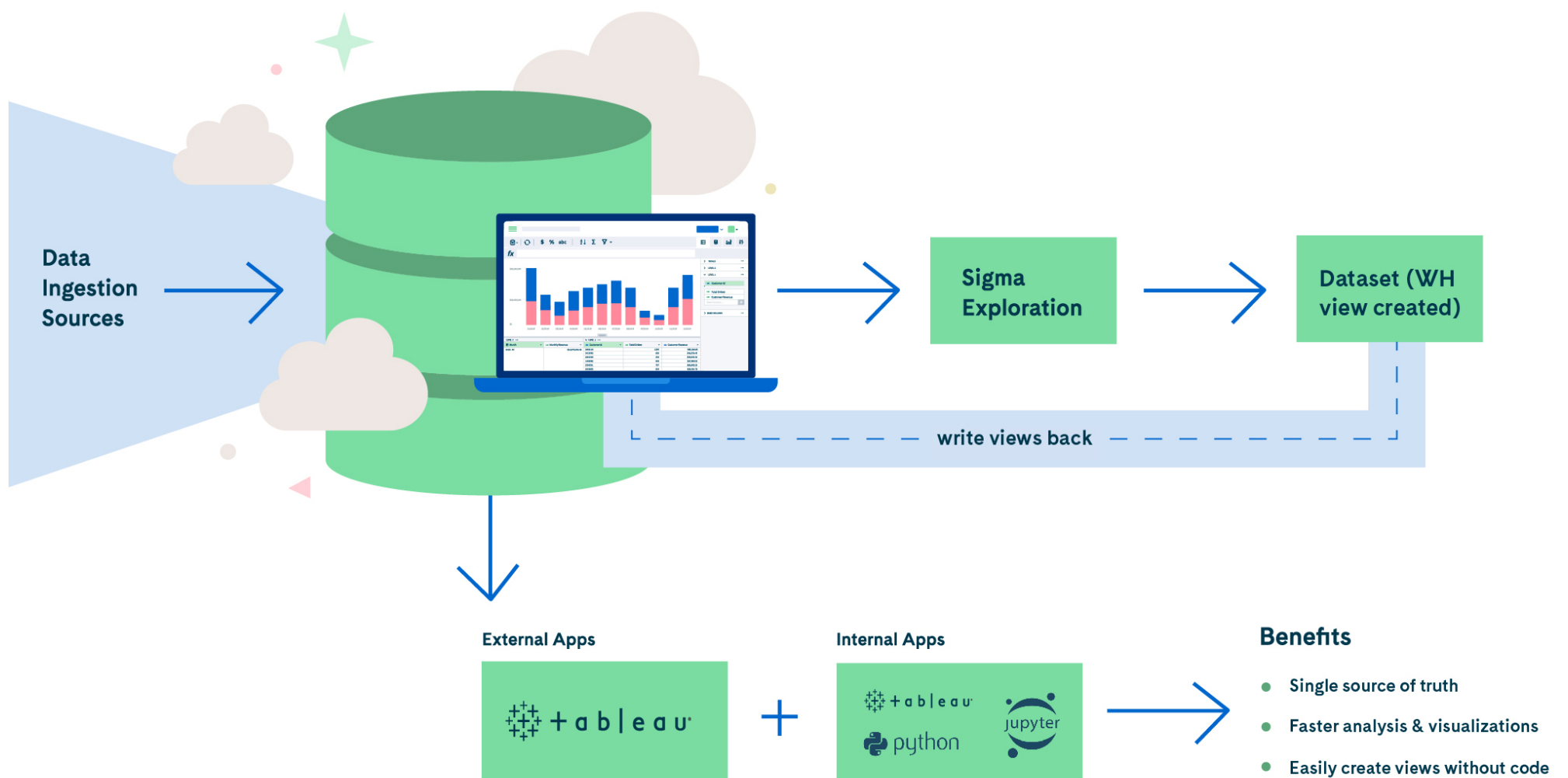
This process benefits both data analysts and domain experts alike. By eliminating the need to write SQL to parse JSON, analysts save valuable time traditionally spent writing code.

For a step-by-step guide to JSON extraction in Sigma, [read our documentation](#).

Write Back JSON Views to Snowflake for Future Analysis

Once you've parsed the JSON into a usable Sigma Dataset, you can write these dataset views back to Snowflake to increase analytic efficiency upstream. Because these Dataset warehouse (WH) views are stored as SQL statements, they remain updated as data changes in the underlying table. Sigma also supports materialization to speed up future queries. Materialization is scheduled to run as needed and stored in Snowflake as tables which can help reduce query times and data warehouse costs — especially when working with datasets that do not update frequently. Dataset views help users unlock the full value of materialization.

These various write back options provide flexibility throughout the analytics lifecycle, giving users additional choices that can help improve analytics efficiency, cut costs, and save time along the way.



Extend the value of Sigma across your data ecosystem by writing Dataset views back to Snowflake and reusing them in internal and external applications.

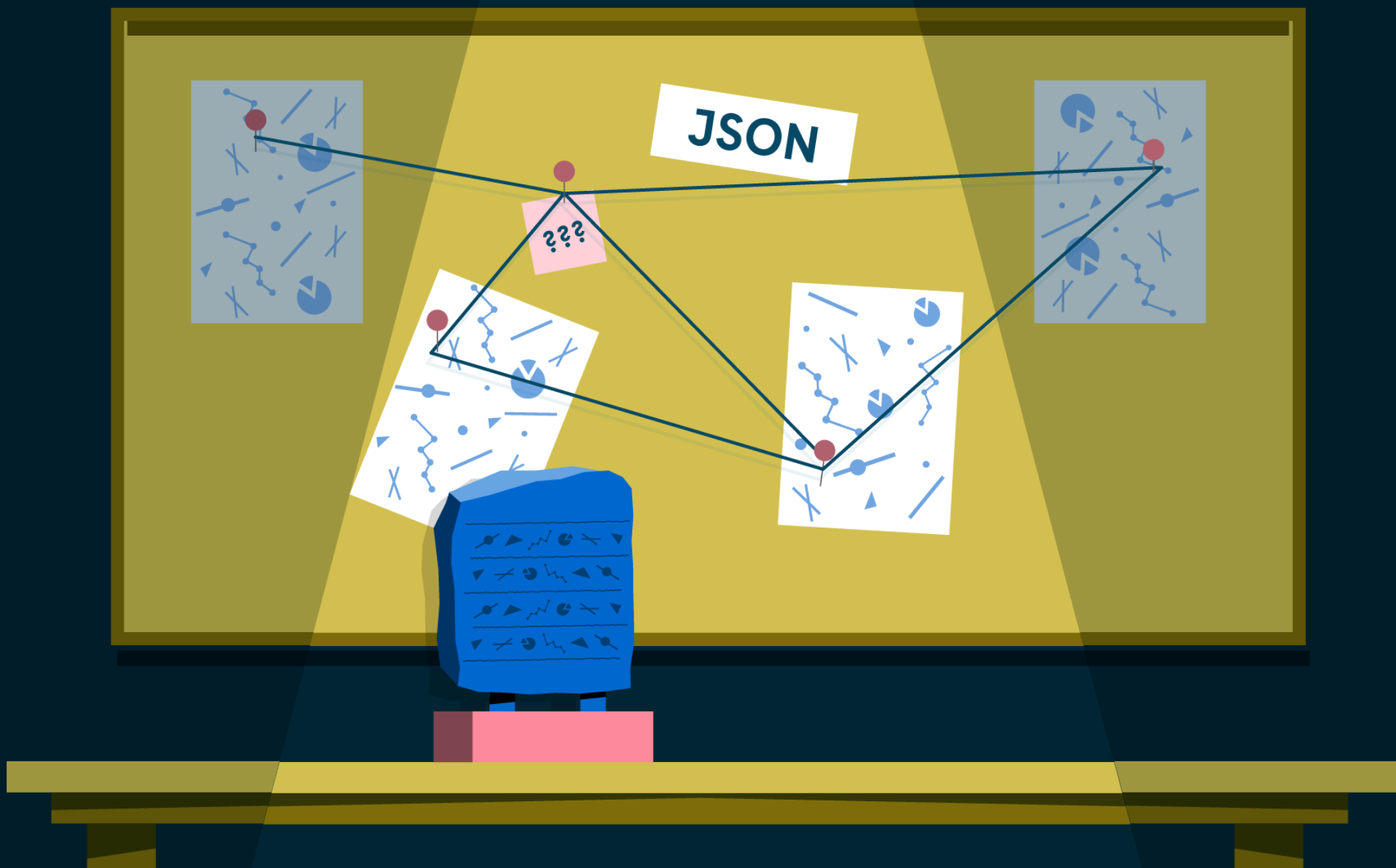
The value of the Dataset views doesn't end in Sigma. JSON views stored in Snowflake can be referenced and reused to power analyses and visualizations across your internal and external data application ecosystem. Sigma customers commonly use dataset views in applications like Jupyter and Tableau. This saves teams valuable time and resources downstream that would be spent unraveling JSON and recreating analyses for future requests.



Sigma makes it easy to extract value from your JSON stored in Snowflake. The spreadsheet interface and write back capabilities simplify data analysis and reduce the time to deliver value from all your data stored in Snowflake.”



Kent Graziano
Chief Technical Evangelist, Snowflake
[Watch the on-demand webinar](#)



Code Cracked

The latest innovations from Snowflake and Sigma have removed the technical barriers to storing, managing, and analyzing JSON. By adopting this combined solution, you can save data experts valuable time and empower business users to unlock the full value of semi-structured data collected from websites, mobile apps, wearable devices, and popular SaaS services.

Without the tools to make sense of JSON, you're not only leaving value on the table, but you're also losing a critical edge over your competition.

To learn more about extracting value from your JSON data, check out our [on-demand webinar](#).

This eBook is brought to you by:



Sigma Computing is changing the way organizations use data. Sigma accelerates time to insight for data stored in cloud data warehouses. Powered by the cloud, the Sigma Spreadsheet, and SQL under the hood, Sigma is a single source of truth that keeps data accurate, secure, and in context. Headquartered in San Francisco, Sigma is democratizing data and leading the community-driven analytics movement. Learn more at www.sigmacomputing.com.



Snowflake's mission is to enable every organization to be data-driven. Its cloud-built data platform makes that possible by delivering instant elasticity, secure data sharing and per-second pricing, across multiple clouds. Snowflake combines the power of data warehousing, the flexibility of big data platforms and the elasticity of the cloud at a fraction of the cost of traditional solutions. Learn more at www.snowflake.com