MANTA

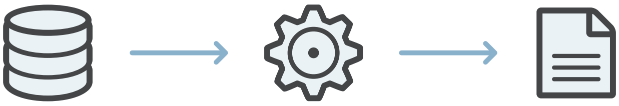
Understand Your Data:

The Ultimate Guide to Data Lineage

The amount of data that companies store and process has skyrocketed over the past few years. Data is becoming more and more complicated and dynamic, and handling it properly and efficiently can sometimes seem impossible. Yes, welcome to the Age of Big Data.

Even though understanding data (where it comes from and how it is linked together) can help companies in many ways, there is still a significant number of enterprises that do not have their data lineage under control.

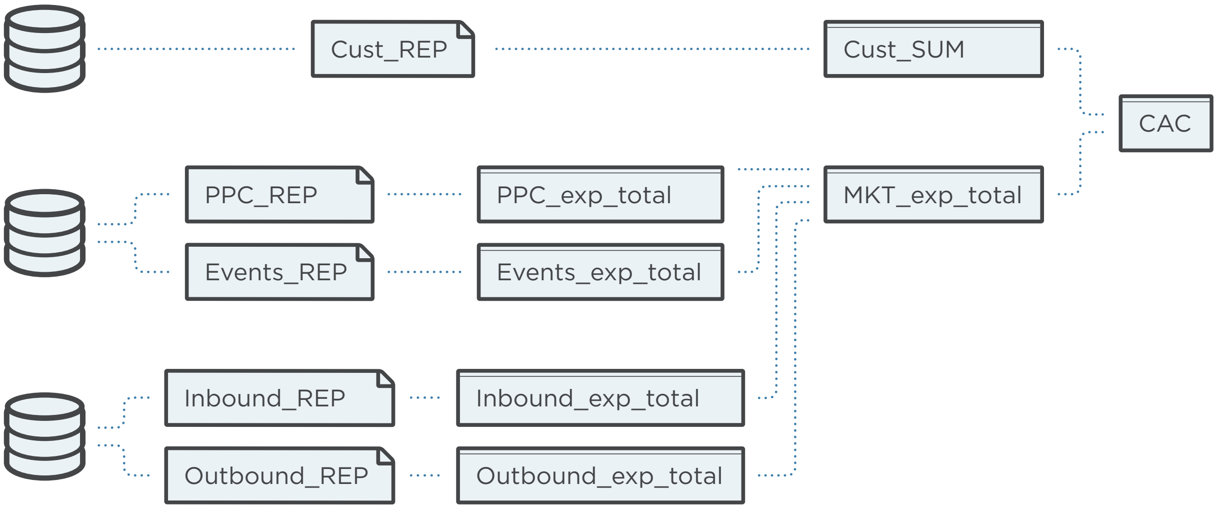
1 What Is Data Lineage?



Data lineage shows what source(s) the data comes from, where is it flowing to in the environment, and—last but not least—what happens to it along the way. This can shed light on the role of particular data units in the environment as a whole.

Let's talk some more about your data. Initially, your data is born somewhere in the environment. It can originate from different databases or be the result of various transformations. This is the beginning of the data life cycle. Data flows in many directions through an environment that typically consists of various platforms, and it passes through multiple processes and storage locations. On its way, the data interacts with other data, is transformed, and is used in different reports.

Just imagine how many systems and sources you have in your organization, how much data processing logic, how many ETL jobs, how many stored procedures, how many lines of programming code, how many reports, how many ad-hoc Excel sheets, etc. It is huge.



Let's take a look at a specific case. The simplified diagram above shows how customer acquisition cost was actually calculated. This metric shows us the expenditures associated with convincing the customer to buy a product or service.

Let's imagine that the final value is the only thing we know. We need to go back and uncover the path that led to this figure. As a result, we will be able to find out what reports or even columns contributed to the final value. In this case, the customer acquisition cost came from a report containing the number of customers and marketing reports with total expenditures for particular marketing activities. We can also see what databases these reports are stored in. What is not obvious at first sight is that procedures are hiding under each line that connects two objects. Knowing all this information allows us to analyze the flow of data, inside and out. It is not necessary to always go backward when exploring lineage.

Sometimes the starting point is somewhere in the middle or at the beginning when you need to make changes to source databases or models. In such cases, being able to track lineage means we can predict what the implications of decisions will be and how they will influence downstream output.

The Role of Metadata

Metadata is basically data about data. It is typically understood as the information about assets and their relationships. When talking about metadata, we typically divide it into the following categories.

TECHNICAL METADATA

Technical metadata, or physical metadata, describes the physical storage of data. It typically covers databases and their schemas, views, and tables, and the columns of those database views and tables. Details for columns also include characteristics such as data type, size, description, and often profiling information. Profiling measures include common value patterns in the data, value frequencies, completeness, and data domains (e.g., a physical column that stores data such as addresses, phone numbers, e-mail addresses, etc.).

LOGICAL METADATA

Logical metadata defines the details that are stored and their relationships to other assets within the system or application. This information is generally created and managed in a data modeling tool. Logical metadata typically covers entities such as customers, parties, and addresses, and becomes the basis for the creation of physical assets where the entities are actually represented.

BUSINESS METADATA

Business metadata has specific meaning with regards to a particular business process—for example, the definition of customer may be unique for each department such as finance, risk, sales, and customer support. The organization needs to define what customer means and determine whether and for what reasons different departments will use different definitions. Business terminology needs to be defined, agreed upon, documented, explained, and tracked along with related processes.

HOW DOES LINEAGE RELATE?

Lineage connects all this information. Asset metadata without lineage is just a definition that is nice to have but without additional value.

Technical vs. business lineage

The requirements for lineage vary depending on who the audience is. **Technical lineage** is a necessity for technical members of the team such as software developers, DBA’s, and report developers who are interested in the inner workings of their code, the movement of data through it, and the impact of their code on upstream and downstream data flows. **Business lineage** allows users to see the journey of their data from a higher perspective, without fine-grained technical details. They often need to know how things are connected, but without having to view detailed transformation syntax. This option is ideal for decision makers who need lineage at a level of detail that increases their trust in the data and the systems they use.

2 Why Is Data Lineage So Important?

What are the consequences of not knowing anything about lineage? Companies who don't care about lineage or are not aware of the aforementioned details are unable to predict what effect their actions may have on their systems. They are unable to equip their development teams with the tools to build new solutions faster and with more accuracy. They are unable to provide their data citizens (data scientists, analysts) with proper information to construct predictive models, and they remain unable to provide their leaders with a trusted platform for making decisions with confidence!

Having a complete understanding of your data, where it comes from, who uses it, and how it is transformed, makes the data trustworthy. With that said, let's take a closer look at the questions that every enterprise needs to be able to answer.

WHAT DATA ARE WE STORING?

This is one of the first things we need to know. This is especially true in data warehouses and data lakes that have evolved over decades, and it is a real problem. The amount of data is astronomical and continues to grow, and the people who built the systems often do not work for the company anymore. This is equally true for disparate transactional systems and archives scattered across the enteprise.

HOW AND WHERE IS IT USED?

What places is the data flowing to? Which jobs are responsible for these moves? This is also an important point of view. The data may reach places and create relationships

ARE WE USING THE DATA AT ALL? IS IT NECESSARY TO STORE IT?

Knowing what data we store enables us to identify the unused parts that died years ago and are now just occupying space we could free up. Time and other resources are also being wasted when we process unnecessary data, a cost that nobody wants to see on their budget. Another situation with similar consequences is duplicate data.

Forbes magazine calls this kind of data dark data and describes it as: "Information assets that an organization collects, processes and stores in the course of its regular business activity, but generally fails to use for other purposes." Lineage analysis helps identify such "islands of data" that are untouched and unused, thus helping organizations avoid wasting valuable resources.

3 What Benefits Does Data Lineage Provide?

Data lineage is your best friend when faced with the following situations and the resulting time pressure.

Compliance

The number of regulations that require data lineage has increased rapidly over the past few years, and we can suppose that there are more of them waiting in line. BASEL, HIPAA, GDPR, CCPA, CCAR... just to name a few. All of these have one thing in common—the company's stakeholders (customers, auditors, employees, control authorities) require accurate tracking of the data we report. Where does it come from? How did it get there? We might know the answers, but are we capable of proving them with up-to-date evidence whenever necessary? Or do we need weeks or months to complete a report which ultimately is not entirely reliable?

Root Cause Analysis

Imagine that you have been working on a project for a long time, but problems occur and your labor does not bear the fruit you expected. You need to find out what happened. Everything seems to be fine as far as your eyes can see. You need to go deeper, under the surface, to find the source of the error, the spoiled root. And this is a challenge often requiring hours and hours of manual labor if automated data lineage is not available.

Impact Analysis

Properly done impact analysis allows businesses to look ahead and determine how changes will impact the organization. When done incorrectly, they can result in delayed deliveries, low-quality deliveries, the need for emergency fixes, and extra work.

Done correctly with unified data lineage, it is much easier to quickly identify the impacts of changes throughout the entire environment. As a result, information about changes can easily be propagated to where it is applicable.

Migrations

Anyone who has ever witnessed a migration project knows how complex a process it is and what amount of labor is necessary to thoroughly scope the project. The project team needs to ensure that the data is secure and that it is also the right time to get rid of the parts that are not needed anymore so they can avoid migrating worthless data. But this is a very tricky task, as it requires the ability to view all dependencies within the environment prior to migration. This can help the project avoid future complications.

Data Consolidation and Virtualization

Data continues to grow and increase in complexity. Many enterprises are consolidating their data from multiple sources in one place or exploring data virtualization technologies that make it appear that the data is in one place. Whether it is being called a data lake, a central repository, or another term is less important in this discussion than being able to identify the original sources of the data and how it arrived at its current location, or where it is really located if data virtualization or replication has been implemented.

3 What can you use data lineage for?

Self-Service Enablement

What is often annoying to scientists or data analysts is that they have to rely on IT to retrieve the data they need. And as you can imagine, this can waste time, delay deliveries, or the data can become outdated by the time it's received by the person who requested it. Armed with the right solution and access to the necessary details surrounding lineage and data origin, data scientists and analysts have the power to retrieve up-to-date information on their own whenever they need it.

Trust in Data and Understanding It

If data lineage is ignored or mapped inaccurately, your decision makers will lose faith in their reports and analytic models. Report developers, data scientists, or data citizens, as they are often called, deserve data that inspires accurate, timely, and confident decision making. Only when you have a complete understanding of your data can you really rely on it and be able to make the most of it, increasing your overall efficiency.

4 Approaches to Data Lineage: How to Create It and Keep It Up to Date

Now that you know what data lineage is, its importance to your company, and its benefits, you are probably wondering how you can actually deliver data lineage.

When you talk about metadata, you very often think of simple things—tables, columns, reports. But lineage metadata is more about logic—instructions, or code, in any form. It can be an SQL script, a database stored procedure, a job in a transformation tool, or a complex macro in an Excel spreadsheet. Data lineage, specifically, can be anything that moves your data from one place to another, transforms it, or modifies it. So, what are your options for outlining, diagraming, and understanding that logic? Here are a variety of approaches that can be taken to achieve data lineage.

Option 1: Pattern-Based Lineage

Solutions exist that estimate lineage information without actually touching or looking at any code. They read metadata about tables, columns, reports, etc. They also profile your data. And then, they use all that information to create lineage based on common patterns or similarities. Tables or columns with similar names and columns with very similar data values are examples of such similarities. And, if you find a lot of them between two columns, you link them together in the data lineage diagram. Vendors might even call it artificial intelligence (AI). There is one big advantage to this approach—if you only watch data, and not algorithms, you do not have to worry about technologies and it is no big deal if a site uses Teradata, Oracle, or MongoDB with Java on top. But this approach is not always accurate. The impact on performance can be significant (you work with data), and data privacy is at risk (you work with data). There are also a lot of details missing (like transformation logic, for example, which is very often needed by your users) and the lineage is typically limited to the database world, ignoring the application part of your environment.

On the other hand, this approach may be sufficient for some cases, especially when reading the logic hidden in your programming code is impossible because the code is unavailable or proprietary and cannot be accessed.

Option 2: Manual Lineage

Manually resolving lineage usually starts from the top by mapping and documenting the knowledge in people's heads. Interviews with application owners, data stewards, and data integration specialists will give you a fair amount of information about the movement of data in your organization. From here, lineage can be defined, usually in spreadsheets or other straightforward mapping mechanisms, to reflect what the subject matter experts have described. Of course, one downside to this approach is that the information may be contradictory, or if you miss talking to someone you simply don't know about, a piece of the flow might be missing! This often results in a dangerous situation where you have lineage but are unable to use it for real case scenarios. The resulting lineage cannot be trusted.

In addition to interviewing application owners and developers, you can also manually review and assess the code itself. Manually examining code, comparing column names, and reviewing tables and file extracts by hand is tedious! It may not even be worth attempting unless you have team members with the requisite skills and expertise in the programs and modules you need to examine. Due to code volumes, complexity, and the rate of change, this method quickly becomes unsustainable. Sooner or later such manually managed lineage will fall out of sync with the actual data transfers in the environment, and once again you will have lineage you cannot actually trust.

Despite these concerns, this approach cannot be sidelined completely, as this is where we all need to start to be able to gain insight into what is actually going on across the entire environment. Sometimes there isn't any code at all or any permissions to access and profile the data directly (especially with legacy systems) and domain experts are your only source of lineage.

Option 3: Lineage by Data Tagging

The idea behind data tagging is that each piece of data being moved or transformed is tagged/labeled by a transformation engine which then tracks that label all the way from start to finish. This approach seems great but typically only works well if there is a consistent transformation engine or a tool controlling every movement of the data. This approach is promising but usually excludes anything that happens outside the walls of the selected engine or technology. Lineage reaches a dead-end because the tags only exist in the closed system. Equally important is realizing that the lineage is only there if the transformation logic is executed. Also, in some systems this method won't be an option because application developers and architects will not want to add formal data columns to the solution model at every touchpoint and for every transfer method applied along the way. One potential solution for complexities with the tagging concept is blockchain, but it is not yet widespread enough to have an impact across the entire data lifecycle in most organizations.

Option 4: Self-Contained Lineage

Some departments have an all-in-one environment providing the necessary processing logic, lineage, master data management, and more. It's all there in one single offering. There are several tools like this, especially with all the new big data / data lake hype. If you have software of this type, it controls everything—every data movement and every change in the data. It is easy for such a tool to track lineage, but still, it remains exclusive to the controlled environment. Your lineage is blind to everything that happens outside the controlled environment. Over time, as new needs appear and new tools are acquired to address them, gaps and dead ends in the lineage start to appear.

Option 5: Lineage by Parsing

Your data lifecycle is complex, heterogeneous, wild, and constantly evolving. The most effective way to manage all your lineage is to do it automatically. This means automatically (programmatically) reading through all the logic and then understanding and reverse engineering it for complete end-to-end tracking. This requires a solution that understands all the programming languages and tools used in your organization for data transformations and movement. And by programming languages, we mean really everything, including graphical flow tools, JAVA, legacy solutions, XML-based solutions, ETL reports, and so much more.

It is difficult to build a solution sufficient to support a single language or tool, let alone dozens of them. Increasing that challenge is the myriad of ways that tools and solutions support dynamic processing. An effective automated lineage solution has to account for input parameters, default values, and runtime information. To effectively automate the delivery of end-to-end lineage to the enterprise, it is critical to parse all these things.

Conclusion

These are the pros and cons of the most productive approaches to data lineage. Find the right balance among them. Ultimately, a combination of such approaches is important. Whether your goals are to achieve better information governance or deliver on-demand lineage for application migration and compliance, look for a flexible lineage solution that offers automated parsing for your most significant technologies and can fine-tune and enhance lineage with manual approaches that are easy to adopt, support, and integrate with your current best practices and existing environment.

WANT TO KNOW MORE ABOUT DATA LINEAGE?

Visit www.getmanta.com/blog to get many interesting articles or drop us a line at manta@getmanta.com.