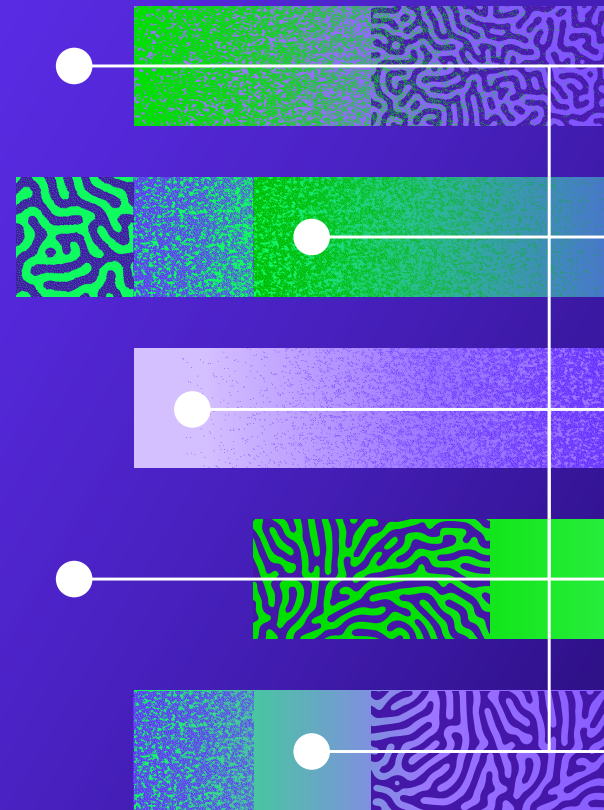




Semantic Layers in Action: Real-World Use Cases and Business Impact

This whitepaper aims to help data leaders understand the importance of having a dedicated semantic layer, its role in the data ecosystem, various use cases and how to maximize the ROI of your modeled data with the dbt Semantic Layer.



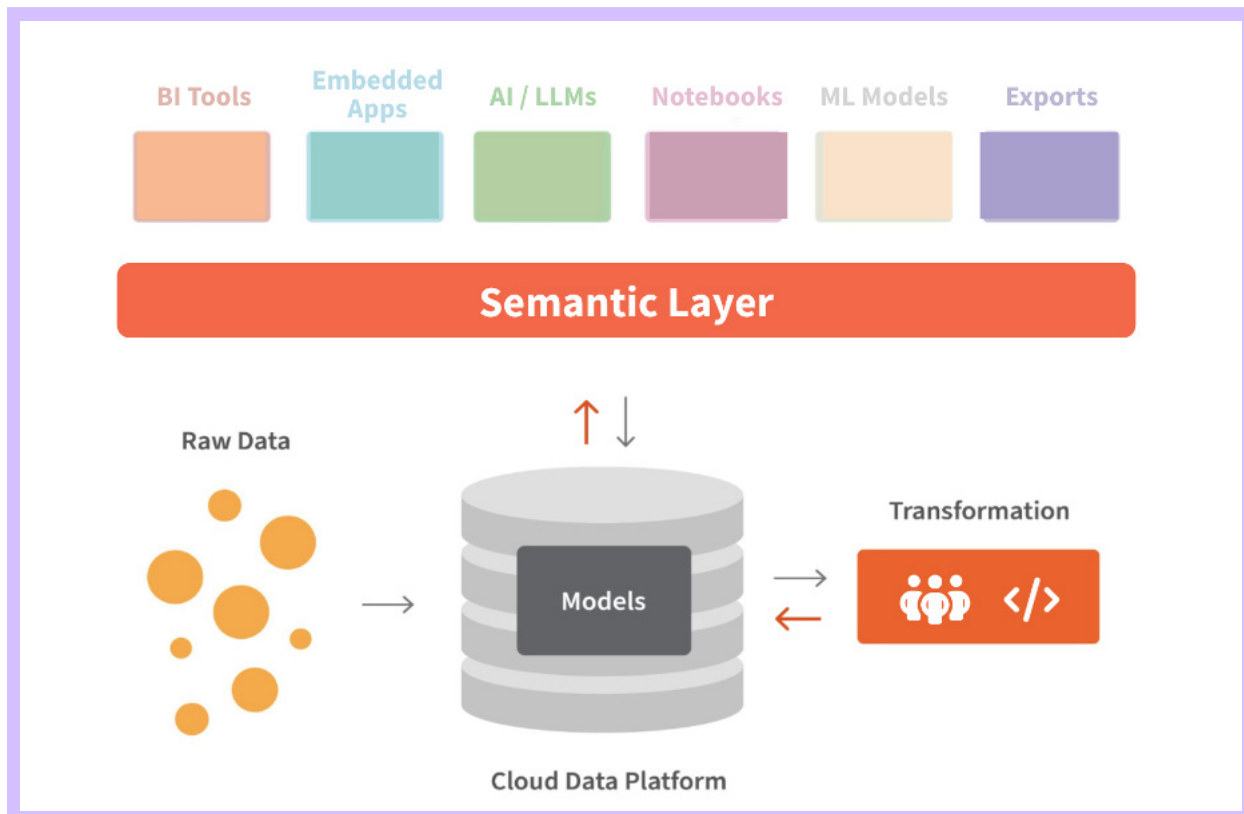
The Challenges of Data Analytics at Scale

Semantic layers address long-standing challenges faced by data practitioners. As organizations attempt to scale data delivery and embrace a data-driven culture, they inevitably face issues related to data trust, data velocity, and costs to maintain and produce insights. Semantic layers provide the foundation to address these challenges through a number of features and use cases.



What Is a Semantic Layer?

A semantic layer is a universal layer of abstraction that enhances data portability, governance and efficiency by standardizing data definitions and business logic across an organization. A semantic layer sits on top of a physical data architecture, providing a single source of truth that can be consumed by various tools and applications.



The [dbt Semantic Layer](#) is one such tool. It is an extension of the data transformations you build in dbt, and it allows you to define and codify your semantic objects on top of your dbt models. It has three key components:

1. The specification allows you to define your [semantic models](#) and [metrics](#). Semantic models are made up of the entities in your data (primary keys), measures (numerical values that can be aggregated) and dimensions (categorical descriptors). Metrics are defined with context on top of these semantic models and are built from measures.
2. The query generation engine, powered by [MetricFlow](#), is a SQL query generation technology that denormalizes data on the fly. MetricFlow is a BSL package that serves as the foundation of the dbt Semantic Layer because it generates the optimized SQL against your data platform to fulfill the request, including handling joins automatically.



3. **The APIs** (GraphQL, JDBC) and **first-class integrations** with tools like Tableau that allow users to consume metrics across a wide array of endpoints (BI tools, APIs, LLMs, etc.)

Why Do You Need a Semantic Layer?

Semantic layers can solve three major pain points for data teams: consistency, governance and cost. Incorporating a semantic layer into your data strategy offers teams a solution to codify business logic, business metrics and complex transformations in one place, making them reusable, consistent and easier to manage across business domains. For data governance, semantic layers enable teams to control the exact parameters around how data is used, aggregated and refreshed where data is being consumed by the enterprise. Semantic layers like the dbt Semantic Layer also provide tools for optimizing query performance by pre-computing common metrics and aggregations.

Customer Quote

“Working backwards, we know the metrics and reports our customers and our internal stakeholders need. They don’t care how that data is built, cleaned or delivered. They just need it. That’s where the dbt Semantic Layer comes in. It gives our data teams a scalable way to provide accurate, governed data that can be accessed in a variety of ways — an API call, a low-code query builder in a spreadsheet or automatically embedded in a personalized in-app experience. Centralizing our metrics in dbt gives our data teams a ton of control and flexibility to define and disseminate data, and our business users and customers are happy to have the data they need, when and where they need it.”

Hans Nelsen (Chief Data Officer, Brightside Health)

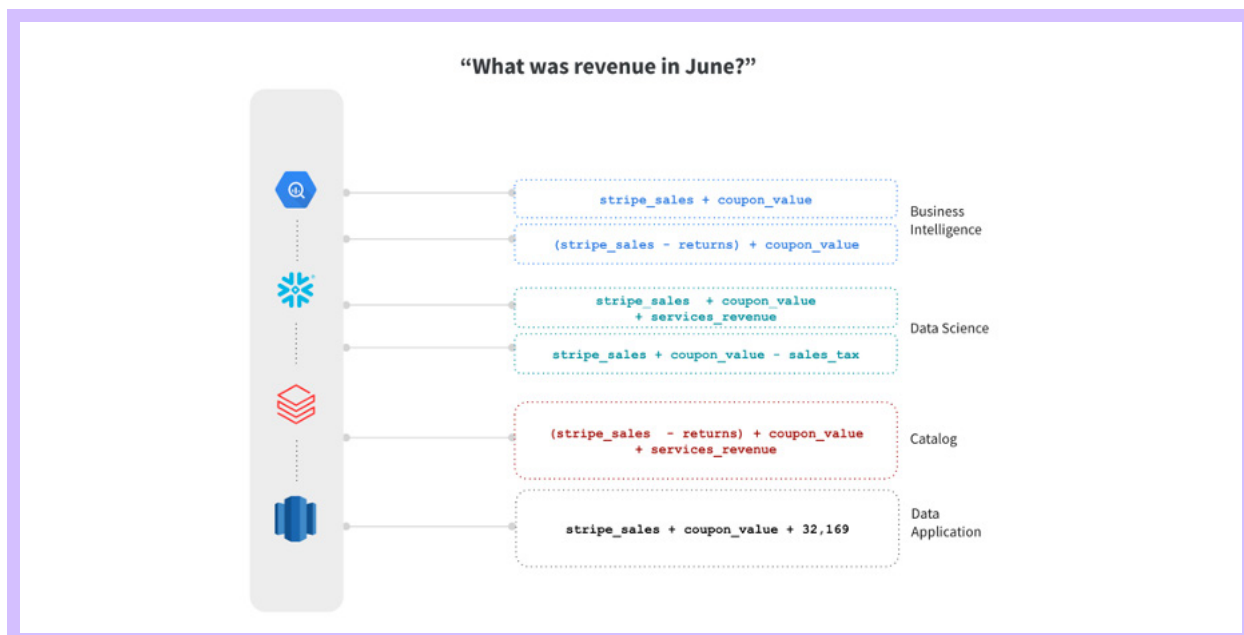
Data Consistency

The core value proposition of a semantic layer is straightforward: It allows practitioners to build, test and govern metrics in a central location so that consistent and trustworthy data with context can be delivered to end-users. These metrics can be delivered in a variety of tools like a spreadsheet, an embedded application or a business intelligence tool like Tableau or Power BI.

Over the past decade, the emergence of the modern data stack has significantly reduced the effort required for building and maintaining production data pipelines. Technologies like Snowflake and Databricks have simplified the complex processes of moving data from source to production, and the volume of data that’s available to be accessed has increased. These enhancements in making data more available have inevitably increased complexity.



While dbt has set a new industry standard for ensuring code quality for data transformation across these ecosystems, helping practitioners manage asset and source dependencies, data accuracy and trust around delivering business metrics to data consumers remains a challenge. Semantic layers are an answer to that issue as they promote data consistency at all reaches of an organization, ensuring end users are not getting different numbers when requesting the same metric. The dbt Semantic Layer fosters this trust and consistency by allowing organizations to codify their metrics in a centralized place (alongside dbt models). This gives users confidence that they're getting the same number and context around that number wherever they consume data.



Semantic Portability

In today's landscape, organizations are often storing and consuming data across a diverse toolset that evolves rapidly, resulting in a number of dependent objects, like dashboards. According to a [Forrester survey](#), on average, a single organization utilizes four or more BI tools (and 25% of organizations use a staggering 10 or more!). In practice, it is very common that different departments in the same organization will rely on different tools, and a universal semantic layer supports that reality without the pain of inconsistency, maintenance costs and vendor-lock in.

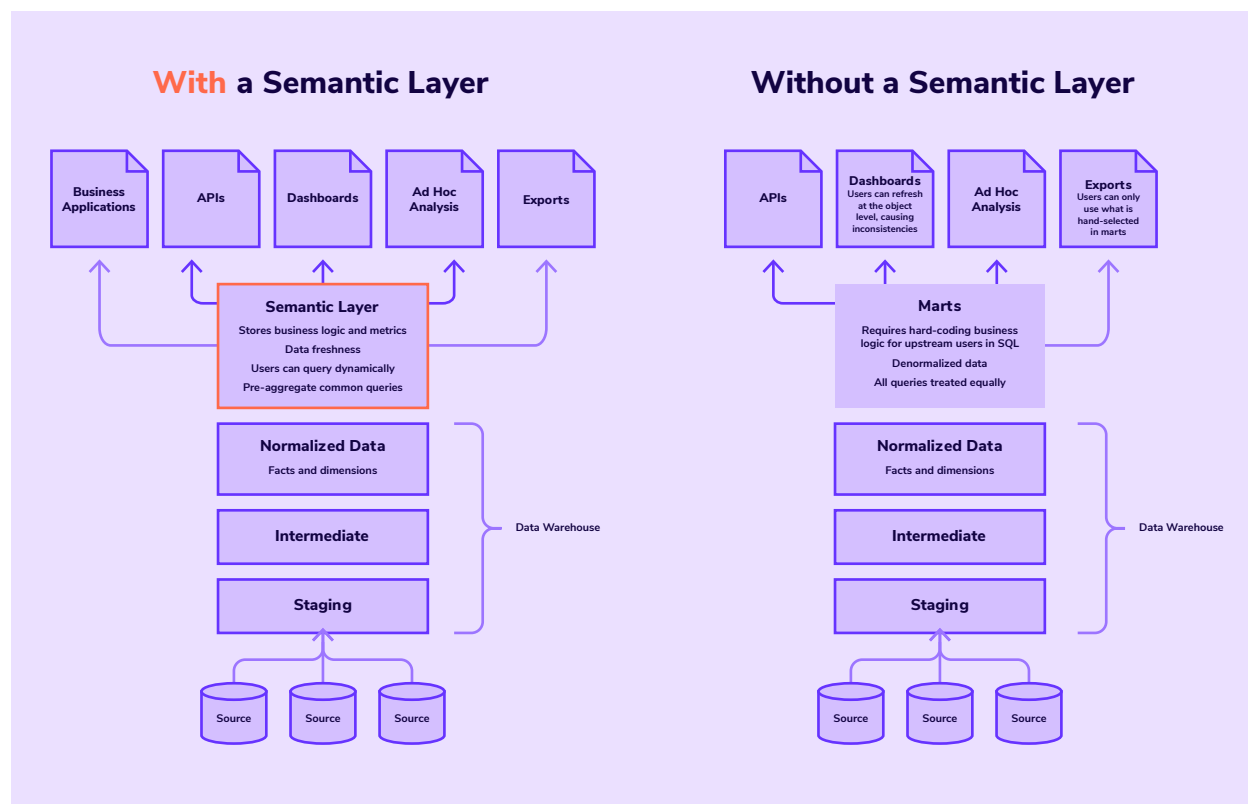
Housing business logic across all of those tools not only degrades data quality, but it's also inefficient. Every time a definition changes, it needs to be modified at each node. It's just a matter of time before definitions diverge.



Semantic layers also play a crucial role in enhancing portability between both upstream tools (data platforms) and downstream tools (including but not limited to BI). They do so by providing a universal layer of abstraction that is accessed and processed in the same way no matter how your data is stored or where your end users are consuming it.

This portability not only allows organizations to meet their end users where they are without sacrificing consistency, it also greatly reduces the time it takes to adopt different tools since the business logic is not tied to any particular one. It allows organizations to avoid vendor lock-in—circumstances that arise where it is simply impractical to migrate off of a given tool due to the effort required.

The dbt Semantic Layer supports these value propositions by remaining agnostic to both upstream tools like data platforms and downstream tools like BI platforms.



Improved Lineage and Governance

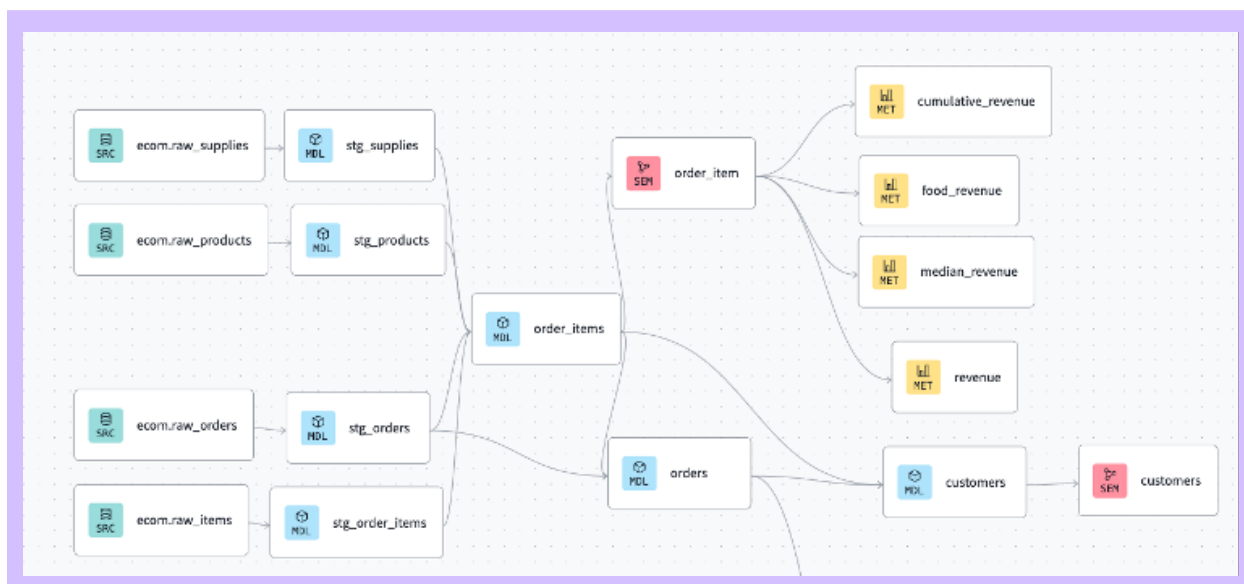
Semantic layers are instrumental in enhancing data lineage and governance capabilities by providing a comprehensive, contextual view of the data pipeline for all types of data consumers.



Using semantic models and metric definitions, data practitioners can clearly define the objects they want to expose to end users, the relationships between those objects and contextual definitions that enrich the objects. Once a semantic model and subsequent metrics are created, they are automatically included in your dbt DAG along with all of your other models, upstream sources and downstream exposures. This is one of the key benefits and differentiators of the dbt Semantic Layer. It's an extension of and tied to your dbt models, where you're already doing transformation, and has incredibly rich lineage from the start of the pipeline (raw data) all the way to the end of the pipeline (metrics and beyond).

Second, the application of version control and [CI/CD](#) within the dbt Semantic Layer ensures that only reviewed code can be pushed into production and faulty code can be rolled back. It also preserves all of the meticulous documentation of model updates and allows source freshness to be observed, fostering transparency and governance around the assets used by downstream tools. In addition, dbt also supports mesh architecture through [dbt Mesh](#). Using dbt Mesh, data teams can define explicit governance rules (through contracts, versions, access levels) for underlying dbt models that make up the dbt Semantic Layer. They can also enable collaborative development with cross-project references and see cross-project lineage in [dbt Explorer](#).

Finally, the dbt Semantic Layer helps data teams build less fragile pipelines by clearly outlining dependencies and making it clear where issues occur. Through these mechanisms, the dbt Semantic Layer empowers organizations to uphold stringent data governance standards, which are crucial for maintaining trust and regulatory compliance.



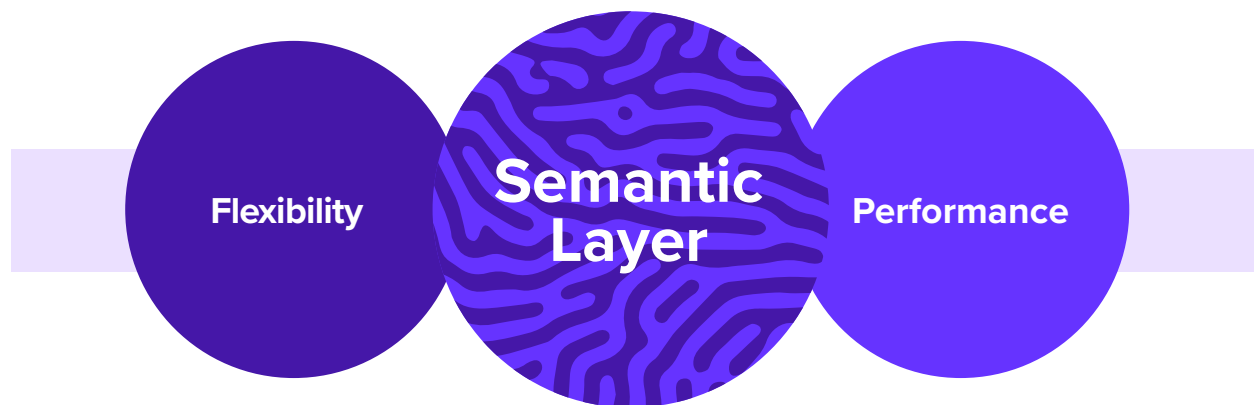


Cost Containment

Another essential feature of the modern semantic layer is the ability to both pre-aggregate and declaratively cache common queries to help optimize performance as well as warehouse and maintenance costs.

One key benefit of the dbt Semantic Layer is that it's dynamic in nature, so the data that is being accessed need not be physically co-located or pre-joined because joins can be generated on the fly. Commonly in data analysis practice, there is a tradeoff between flexibility and performance.

The dbt Semantic Layer offers the best of both worlds. For some common requests powering an executive dashboard (as an example), it makes sense to have those results pre-aggregated and cached rather than computed on the fly for faster queries and cost containment, and data producers can use the [declarative caching feature](#) in the dbt Semantic Layer to do this. Caching is straightforward as it is simply a configuration that can be made in the dbt project configuration for any combination of objects. Additionally, the cached tables are managed and invalidated as needed by the dbt Semantic Layer, so data teams do not need to maintain them.



Caching allows the dbt Semantic Layer to offer the best of both worlds for flexibility and performance. Data teams can cache common requests so they don't incur on-the-fly compute costs for common queries while their end users have the flexibility to discover and query less-used data dynamically. They can do so without having to spend resources on pre-computing or asking a member of the data team to create a view to answer their question.

The dbt Semantic Layer also supports [Exports](#), which allow you to write data back directly to your data platform as standalone tables (separated from the dbt Semantic Layer). Exports are run with your dbt projects, so data is always up to date. Exports are

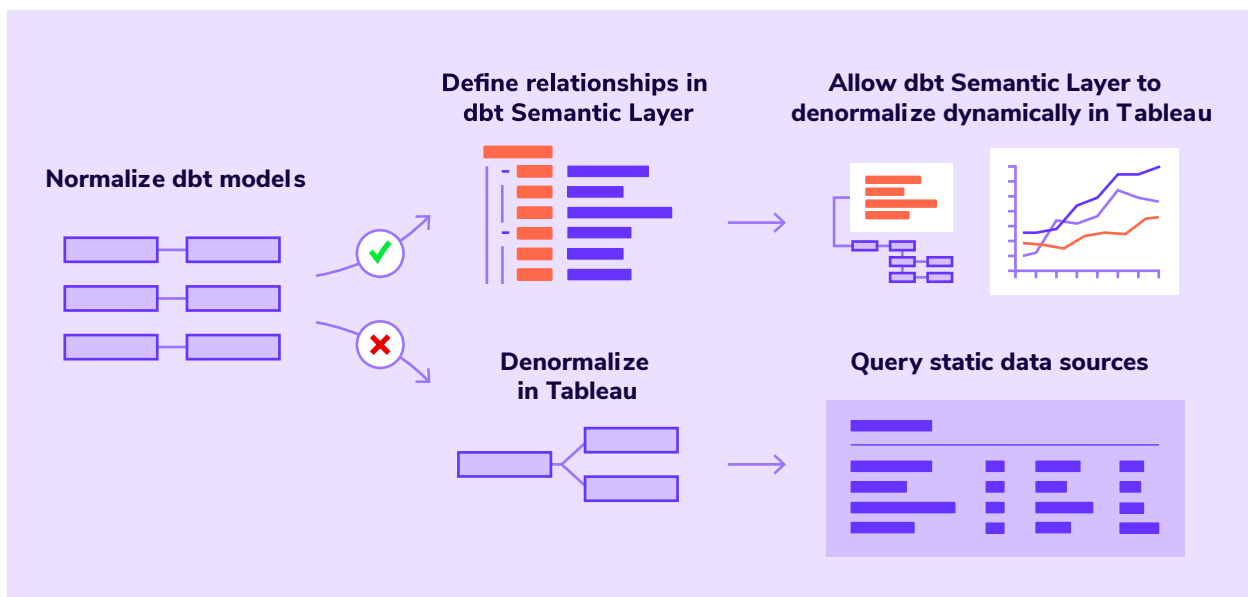


not dynamic, but they are a very useful means to use the semantic layer in a tool where a first-class integration does not yet exist. Exports are also a powerful starting point to adopt the semantic layer because data teams can continue to consume information in the same way they are used to (from tables) while improving data quality and greatly reducing maintenance costs. Users don't have to manage building new tables from scratch, maintain static tables of data or write complex SQL. Instead, they can programmatically denormalize their data.

Technical Overhead Reduction

Semantic layers give data teams a path to reduce the overall complexity and number of data assets in their environment. Traditionally, organizations would create a layer of domain-specific, denormalized views or tables that were purpose-built for analytics. Each of these assets required gathering requirements, writing SQL and additional testing. Moreover, if any of these assets required changes (e.g., another dimension or a different time grain), it's time consuming and error prone to make and maintain them. Alternatively, organizations may also build metrics reporting into the BI reporting layer, which leads to inconsistencies and does not allow for the ability to version control logic and lineage of the data. Neither of these options set organizations up to properly scale their data.

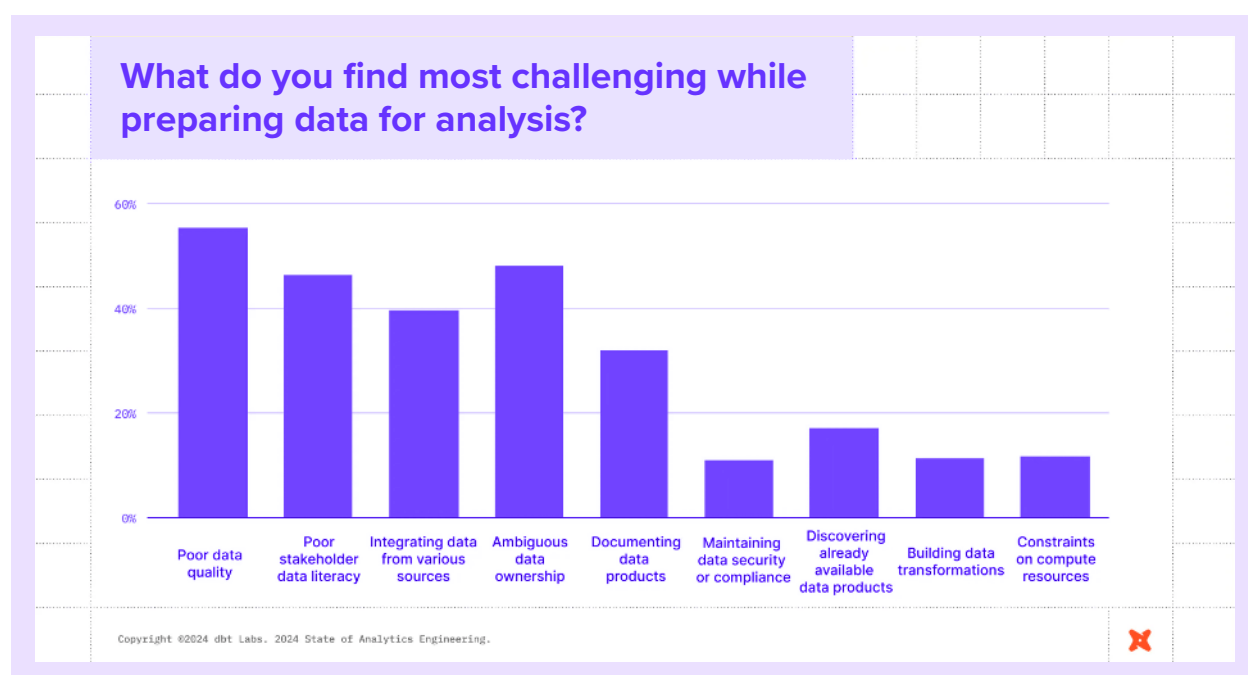
The dbt Semantic Layer works to reduce technical overhead by eliminating the need to write new domain-specific models. Instead, semantic models can be built using YAML like the rest of your dbt project, where dimensions, metrics, filters and time-grain are easily managed, and metadata can be captured for each of these elements. Once this is created, all of these assets will appear in your dbt project docs and your DAG, and they are subjected to the same governance processes as other assets.





Driving Culture Change

Semantic layers compel a vital culture change within an organization by addressing the pervasive issue of data silos. Silos occur when departments or teams operate in isolation, each utilizing disparate data definitions and tools, often leading to inconsistent results and misaligned insights. Individuals working within these silos may produce varied interpretations and analyses of the same data, resulting in roadblocks that can get in the way of decision-making and strategic alignment. Additionally, a recent finding from one of our [“State of Analytics Engineering”](#) reports showed that poor data quality and poor stakeholder data literacy are among the most commonly cited challenges for data practitioners today.



The implementation of a semantic layer standardizes data definitions and business logic across the organization, ensuring that all teams operate from a single source of truth. This forces critical conversations about how metrics should be tracked and measured, fostering a collaborative environment where stakeholders agree on canonical definitions. By aligning on these standards, organizations not only enhance the accuracy and reliability of their data but also create a more cohesive data-driven culture.

Through many of its features, the dbt Semantic Layer helps support culture change, specifically by codifying key metrics and supporting the various metadata around them. This is a helpful forcing function that brings organizations together to agree on the canonical definition of key components.”



Use Cases for a Semantic Layer

AI Readiness

Another emerging pattern for the semantic layer is its significant contribution to enhancing AI readiness and improving the accuracy of Large Language Models (LLMs) by providing meaningful, structured, well-defined data. This ensures that LLMs receive high-quality, context-rich metadata, as well as guidelines about how data ought to be used in the business domain, thus leading to more reliable results. As discussed in depth by the data.world co-founders in their [groundbreaking research paper](#), the semantic layer organizes data into meaningful models for different domains of the business with their own purpose and context.

By enriching your LLMs with high-context, well-governed data inputs, you can ensure high quality outputs across your AI stack. Using the dbt Semantic Layer, you can set up your AI investments for success:

- **Reduce hallucinations:** Get high-quality AI responses backed by real data, which keeps your models on track and can be enriched by business context
- **Build once, use everywhere:** Define your business logic via metrics in the semantic layer and access it through any connected LLM.
- **Lower the barrier to analytics:** Democratize data-driven decision-making by empowering less technical users to “self-service” the answers to their questions using an agent that utilizes natural language.

As an example of what can be accomplished by powering your LLM with governed semantic definitions, dbt Labs built an agent to interact with the dbt Semantic Layer using plain language text: Ask dbt. Unlike traditional AI chatbots, Ask dbt uses the dbt Semantic Layer to provide critical context about your dbt project, improving accuracy by 3x [as observed in a dbt Labs benchmark](#) study. With Ask dbt, users can ask questions in natural language and receive insights in an understandable format, which can significantly speed up business processes and decision-making.

The screenshot shows a web interface for 'Ask dbt'. The user has entered the query 'break down revenue by product'. The interface shows the query being processed and the results displayed in a table format.

PRODUCT_PRODUCT_NAME	REVENUE
chai and mighty	13298
cat code challenge	48408
cat visit 1 hour	39288
cat ai food prediction	39688
cat butler	154688
cat visit private	355148
tangaroo	17788



BI / Reporting

Semantic layers bring teams on the same page with contextualized, codified metrics, ensuring everyone is always looking at the same number. Organizations are always in need of populated dashboards and reports about the performance of their business. Without a semantic layer, there are a few patterns we observe that ultimately lead to distrust in data, delivery bottlenecks and high costs to produce an insight:

Metrics Defined in Respective BI Tools

Without a governed source of truth, there are bound to be inaccuracies, misalignment and high costs to make updates if a particular metric changes. Notably, organizations will spend more time determining what the correct number is rather than making a decision for the business based on the data.

“Meeting end users where they are” is a key tenet of the dbt Semantic Layer, which supports a number of [first-class integrations](#) with consumption tools as well as generic APIs that can be used to create custom applications to deliver your data.

Maintaining Denormalized Metric Models in Your Data Platform

Data teams must spend time maintaining these tables and using warehouse compute to populate them, even if only a small percentage of that data is used. This is a high cost of maintenance because metrics are seldom useful on their own. They require dimensions (categorical attributes of data like “country” or “product name” alongside revenue) that business users can use to further slice data and answer their questions. If you create a table with a metric, you’ll need to create numerous other tables derived from that table to show the desired metric sliced by the desired dimensions and attributes. Mature data models have thousands of dimensions, so this will add up quickly and lead to a lot of duplication and added cost.

In addition to helping data consumers, a key purpose of the dbt Semantic Layer is to reduce the cost of producing insights by centralizing governed metrics, reducing duplication, enabling self-service and helping your data teams focus on more high-value work.



External Analytics and Embedded Applications

Part of the benefit of a universal semantic layer is that it can plug in anywhere. External analytics involves organizations exposing data to external partners or customers. This use case in particular greatly benefits from the implementation of a semantic layer due to the risk, high value and sensitivity of the data being shared. When data is exposed externally, it is often tied to monetized applications and revenue-generating use cases, making accuracy, consistency and security paramount.

The dbt Semantic Layer provides generic, user-friendly APIs like [JDBC](#) and [GraphQL](#) to help serve these external stakeholders in a custom capacity. Developers can build custom web apps using these APIs to serve up relevant, personalized data to end-users without driving up costs with legacy BI tools. This consistency is crucial for maintaining trust and reliability in external analytics applications.

Self-Serve Analytics

Semantic layers offer a pragmatic solution to the daunting mission of providing an interface for business users to access analytics in a self-service manner. In dbt, semantic models infer relationships between models in your project to create a single data source that is representative of a business process or domain that can be consumed virtually anywhere. This becomes especially powerful if you have a distributed data workforce using spreadsheets, data exports and BI tools who need to tap into a common definition of truth.

The dbt Semantic Layer offers integrations with popular business applications like [Google Sheets](#) and [Excel](#), so you can introduce your production data models into the lowest common denominator of your workforce. By empowering business users to access, analyze and generate insights from data without requiring deep technical knowledge, organizations can drive efficiency and innovation. This self-serve capability helps democratize data, enabling more users to contribute to data-driven decision-making and reducing the bottleneck on data teams.

METRIC_TIME_MONTH	PRODUCT_PRODUCT_NAME	REVENUE
2016-06-01T00:00:00.000Z	vanilla ice	1998
2016-06-01T00:00:00.000Z	chai and mighty	1405
2016-06-01T00:00:00.000Z	nutellaphone who dis?	1639
2016-06-01T00:00:00.000Z	ml-bun	1644
2016-06-01T00:00:00.000Z	flame impola	1950
2016-06-01T00:00:00.000Z	tangaroo	1944
2016-06-01T00:00:00.000Z	adobe-wifi	1346
2016-06-01T00:00:00.000Z	for richer or pourover	2386
2016-06-01T00:00:00.000Z	the keyboard	1644
2016-06-01T00:00:00.000Z	doctor shoe	1964
2016-10-01T00:00:00.000Z	tangaroo	2262
2016-10-01T00:00:00.000Z	flame impola	2324
2016-10-01T00:00:00.000Z	chai and mighty	1890
2016-10-01T00:00:00.000Z	nutellaphone who dis?	1903
2016-10-01T00:00:00.000Z	doctor shoe	1969
2016-11-01T00:00:00.000Z	for richer or pourover	2493
2016-11-01T00:00:00.000Z	flame impola	3066
2016-11-01T00:00:00.000Z	doctor shoe	2191
2016-11-01T00:00:00.000Z	the keyboard	2349
2016-12-01T00:00:00.000Z	vanilla ice	3702
2016-12-01T00:00:00.000Z	for richer or pourover	4263
2016-12-01T00:00:00.000Z	nutellaphone who dis?	3443



Furthermore, AI readiness also ties into the self-service use case. By using the dbt Semantic Layer to confidently generate insights using simpler and less-technical interfaces (like the Askdbt bot), a greater number of business consumers can truly be self-sufficient.

Exploratory Analytics

Exploratory analytics is essential in the data science process, using Python libraries to explore data, detect patterns and test hypotheses that drive successful business strategies and enhance data ROI.

However, exploratory analytics can be complex. The constant influx of data sources requires integration from multiple origins to form a comprehensive view while ensuring data quality and integrity. This expense can hinder productivity, making it crucial for data teams to have the right tools and resources to make the iterative process of exploratory analytics effective and goal-oriented.

The adage “garbage in, garbage out” applies to training ML models as well. Data scientists need governed metrics definitions to ensure their datasets accurately represent the business metrics they aim to enhance. Without this confidence, their data modeling efforts are pointless.

The dbt Semantic Layer empowers data science teams with centralized, governed metrics that coexist with other data models, supporting exploratory analytics workflows. Metrics and models are version-controlled, allowing for full traceability from source to metric, and teams can easily merge semantic models with other dbt models. With built-in integrations to tools like Hex and Mode, data can be exported back to the warehouse as centralized, governed metrics tables. It can also be queried via the [JDBC API](#) , [GraphQL API](#) or [Python SDK](#) in the notebook of their choice.



Getting Started with the dbt Semantic Layer

Semantic layers are a vital part of a modern data stack and are important in bringing everyone in an organization on the same page while simultaneously improving workflows for practitioners who are delivering data.

Through various implications across AI, BI, embedded analytics, self-serve and exploratory use cases, the dbt Semantic Layer can help organizations become truly data-driven. It does so by providing consistency to key metrics across an organization, reducing the cost of producing insights and empowering more stakeholders to participate in the data lifecycle.

Learn More

dbt Cloud Teams and Enterprise plan customers can get started with the dbt Semantic Layer with up to 1,000 queries for free per month. Here are some other resources to learn more and get started:



[Discover more dbt Semantic Layer features and use cases](#)



[Get started by reading the dbt Semantic Layer docs](#)



[Join a live dbt Semantic Layer workshop](#)



[Get dbt Semantic Layer integration help from InterWorks](#)



About the Authors



Roxi Dahlke | Staff Product Manager, dbt Labs

Roxi is a product manager for the dbt Semantic Layer. She was a pioneering member of the product team at Transform, where she played a key role in building the product prior to its acquisition by dbt Labs. Roxi's career has consistently centered on bridging the gap between data and people. She is based in San Francisco, California.



Jack Hulbert | Analytics Architect, InterWorks

For the last decade, Jack has worked as a solution and product architect for startups and data consultancies, where he has worked with some of the world's most admired companies on their data strategies. He is based in San Francisco, California.

About InterWorks

[InterWorks](#) is a people-focused tech consultancy delivering premier service and expertise in collaboration with strategic partners. Our clients trust us to guide them through their unique challenges, bringing together the right people and technologies to help everyone succeed. Covering data strategy, architecture, management, visualization, AI, ML and more, we support you in every aspect of your BI strategy. We value laying a strong foundation, and our relational approach allows us to customize solutions that empower you to do data your way.

About dbt Labs

Since 2016, [dbt Labs](#) has been on a mission to help analysts create and disseminate organizational knowledge. dbt Labs pioneered the practice of analytics engineering, built the primary tool in the analytics engineering toolbox and has been fortunate enough to see a fantastic community coalesce to help push the boundaries of the analytics engineering workflow. Today there are 40,000 companies using dbt every week.