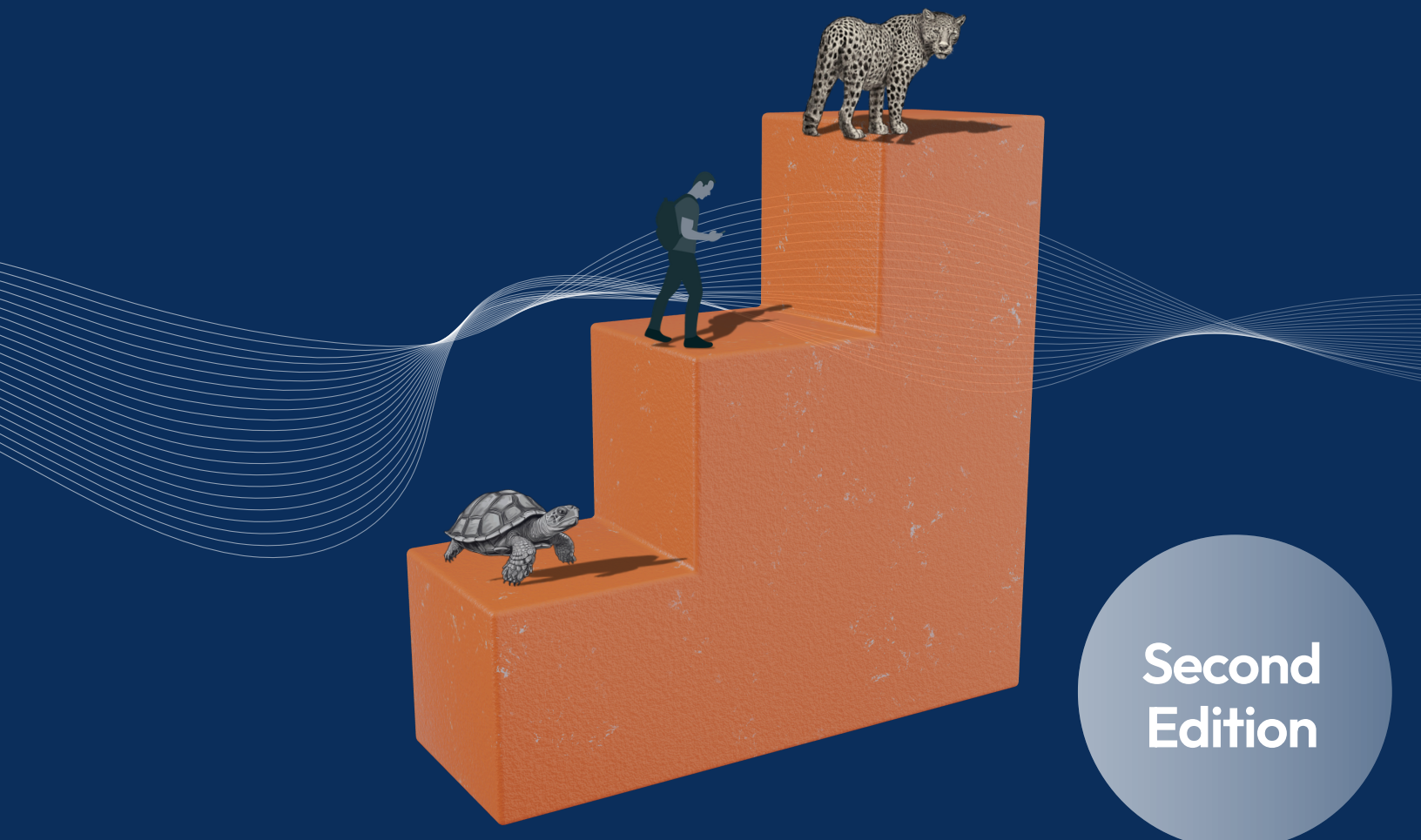
2026

MONTE CARLO

The Data Quality Maturity Curve

Learn how to crawl, walk, and run in the age of AI.



Second Edition

Important disclaimers

- This guide will reference a variety of numbers as benchmarks. Remember that these numbers are only intended as guidelines and will vary based on your organization's specific context, industry, and data maturity.

- Even with a smaller number of tables or columns, if the complexity or criticality of the data is high, your organization might opt to implement advanced practices earlier.

- The quality of data is not just about volume; it's about how critical that data is to your organization's operations, decisions, and strategy. Hence, always consider the business context alongside these numbers.

TABLE OF CONTENTS

What does your data quality really need?

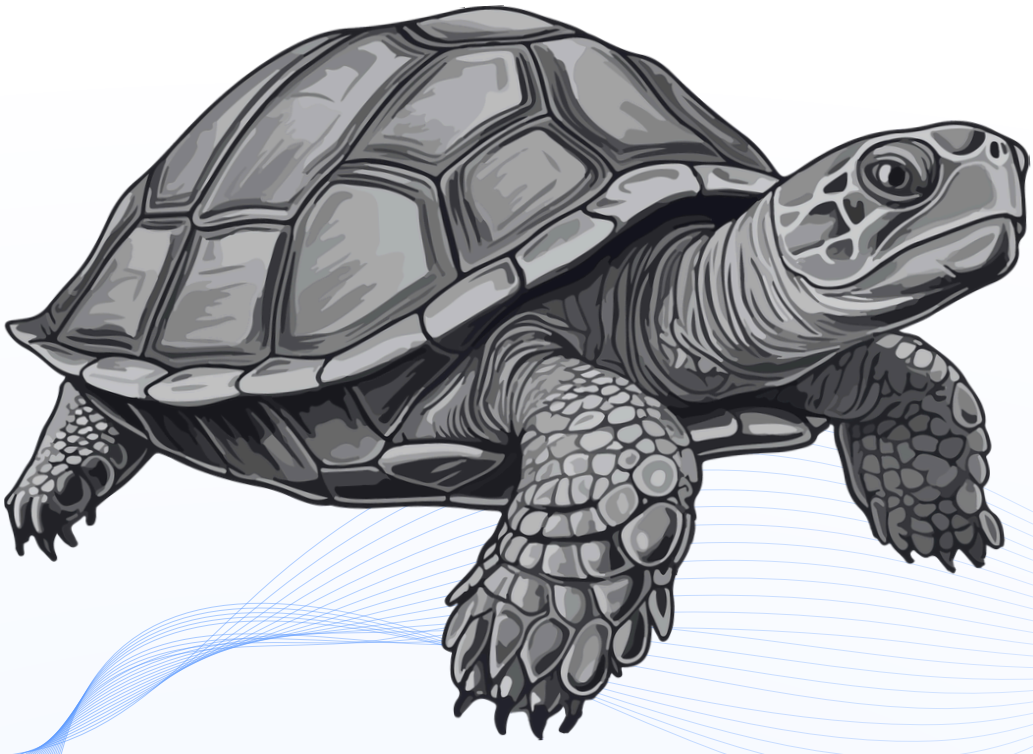
Data quality is important, there's no doubt about it. But with the specter of a world dominated by enterprise AI lurking just around the corner, bad data isn't just an inconvenience; it's a zero-day liability for executives.

For the first time in tech's history, the quality of enterprise data isn't just subtext – it's the main headline. But, as data quality finally takes its rightful seat at the platform table, where should new data teams start?

Like any new endeavor, developing a scalable data quality strategy doesn't happen overnight. We grow into it. Selecting the right data quality solution isn't about developing the ultimate strategy for forever. It's about developing the right strategy for right now.

In this guide, we examine the **Data Quality Maturity Curve** – a representation of how data quality is monitored and evaluated at different stages of your organizational and analytical maturity – to offer some experienced perspectives on where you should be at each point in your data quality journey.

Let's get moving.



01 The Data Quality Maturity Curve

- Data observability
- SLAs
- Governance
- Data contracts
- Deep monitors

- Data observability
- Automated monitors

- Manual review
- dbt Tests

The Data Quality Maturity Curve

Data quality is defined by how accurate, reliable, complete, discoverable, trustworthy, and actionable a specific dataset is for a given use-case.

The fundamentals of data quality necessitate that:

- **The data is current, accurate, and complete**
- **The data is unique and free from duplicates**
- **The model is sound and represents reality**
- **The transformed data is free from anomalies**

As stakeholders clamor for more and more data, it's easy to secure investments for new pipelines, better discovery tools, and new ways of manipulating and visualizing the data. But all that investment will be for naught if we don't equally invest in the right data quality coverage to deliver the projected ROI.

You can think of the data quality maturity curve in terms of the "crawl, walk, run" mentality. You can't run before you crawl. However, over time, you'll become more sophisticated. Your data will grow. Your platform will expand. Your team will mature. And as you avail yourself of new resources and use cases to meet your growing data needs, your data quality system will need to grow alongside that added complexity.

This framework will help you understand where you're at – and where you should be – in your data quality journey relative to the growth stage of both your organizational maturity and your platform.

So what does it mean to crawl, walk, and run when we talk about data quality maturity?

Let's explore.

02 Crawling with manual tools

```sql
-- tests/order_value_test.sql

SELECT order_id, quantity, price_per_item,
total_order_value

FROM {{ ref('orders') }}

WHERE total_order_value != ROUND(quantity *
price_per_item, 2)
```

Crawling with manual tools

When you're just getting started on your data quality maturity journey, it's unlikely your organization will need an ultra-sophisticated data quality solution. If your data volumes and variety are relatively small, and data use-cases are limited, you can likely survive by just manually inspecting the data or adding a few dbt tests.

Below is a breakdown of when that might make sense.

Manual Inspection

- Tables: 1-10
- Columns per Table: 1-30
- Rows: Up to several thousand rows per table

With a limited dataset, data teams can manually inspect tables and columns by simply scanning the data to identify inconsistencies, missing values, and a variety of other data quality issues.

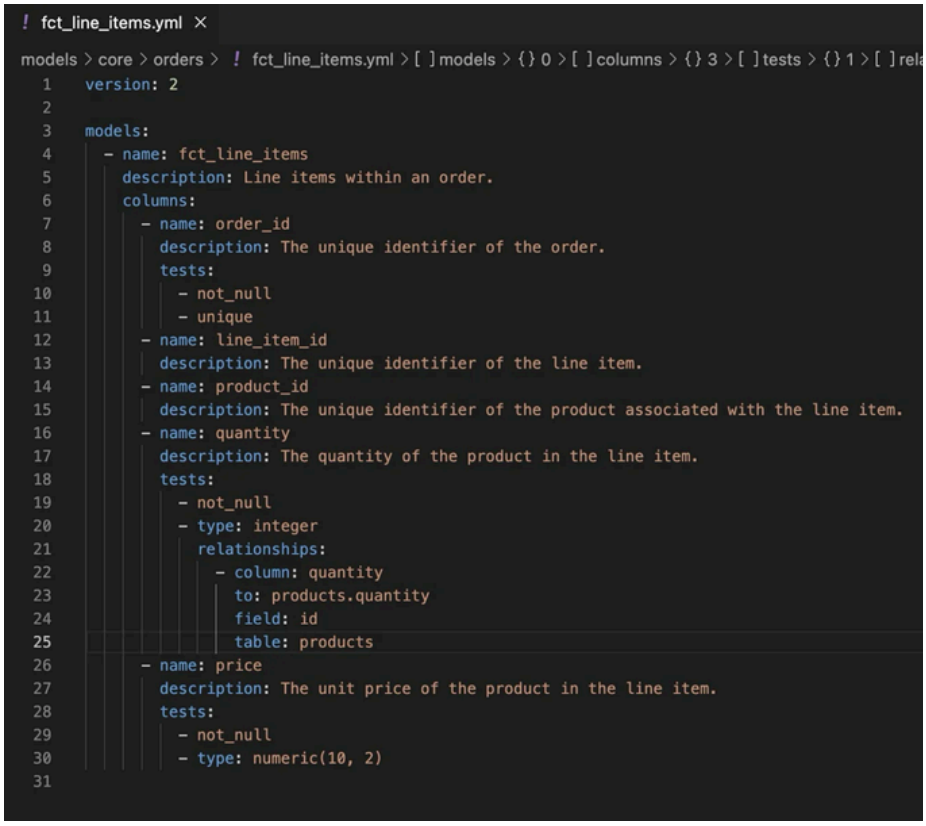
Unfortunately, this strategy will quickly become unmanageable as your data ecosystem grows and you continue to rely on human efficiency to effectively monitor what's likely to become thousands of rows. For this reason, you'll want to start doing some form of simple testing as soon as you can.

Transition to dbt

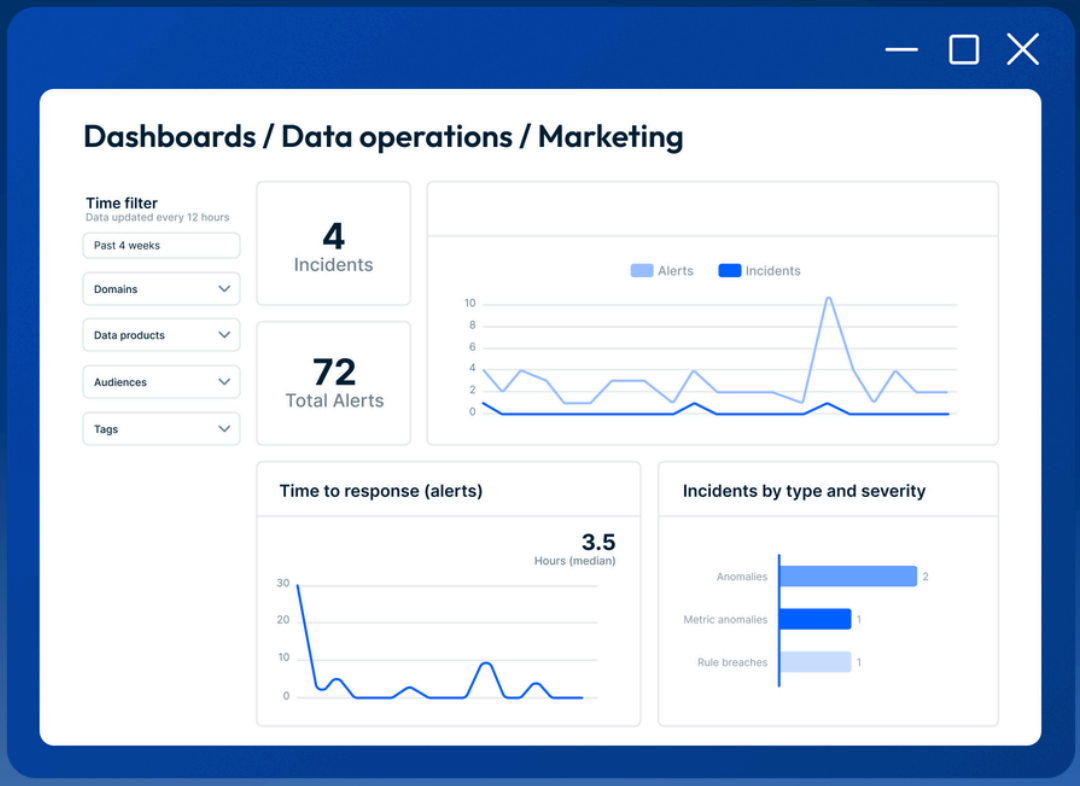
- Tables: 10-50
- Columns per Table: 30-50
- Rows: Hundreds of thousands to a few million rows in total, across all tables

As your datasets start to grow beyond what can be manually inspected, there's a transition to using dbt for more structured data testing. Column-level tests are manually added to validate data integrity and ensure specific data quality standards.

With dbt you will also have table level ineage and you can start to capture documentation – which is a good habit to get into regardless of where you're at on your data quality maturity journey.



03 Walking with data observability



Walking with data observability

Crawling is your first venture into data quality, but walking is where you start building for the future.

In the same way that walking is the foundation of running, **the second stage of your data quality journey is all about scalability**. And this is where automation comes into play in a big way.

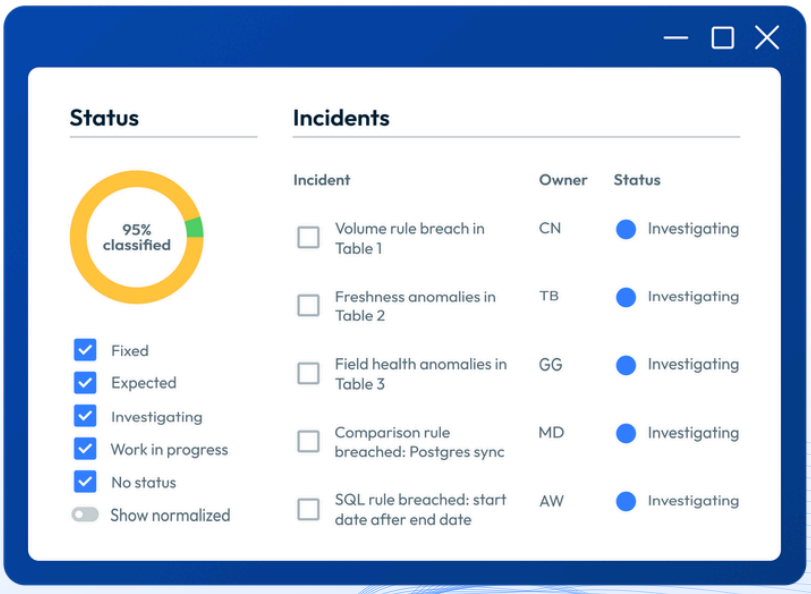
As the number of tables and columns grows, and the volume of data increases, manual methods become less efficient. And as the diversity of data and its sources evolves, you'll need better tooling to keep up.

Here's what that might look like:

- **Tables:** 50-200
- **Columns per Table:** 50-200
- **Rows:** Several million to a billion rows in total

At this point, scale and operational efficiency are the priorities for your data quality. In order to ensure you continue to feed fresh, reliable, and accurate data into your pipelines, you'll need an automated end-to-end solution to scale your data quality across your growing data environment. And this is where data observability solutions like Monte Carlo come into play.

Let's look at some of the key benefits of data observability and how it helps teams take their data quality from crawling to walking with confidence.



Automation

Automation is truly the secret sauce for scaling data quality. Unlike the varying degrees of manual monitoring we looked at in the previous section, data observability solution Monte Carlo provides automated detection and alerting for data anomalies right out of the box, including automatic thresholding based on metadata and the ability to quickly create deep monitors for known issues and critical SLAs.

This means that your data engineers can programmatically scale data quality in tandem with the scale of their pipelines.

It also means that, as your platforms become increasingly complex, your data quality will expand to meet the needs of your evolving platform layers.

A good data observability solution will also include automation for another critical element that was lacking in the crawl phase: column-level lineage.

Column-level lineage

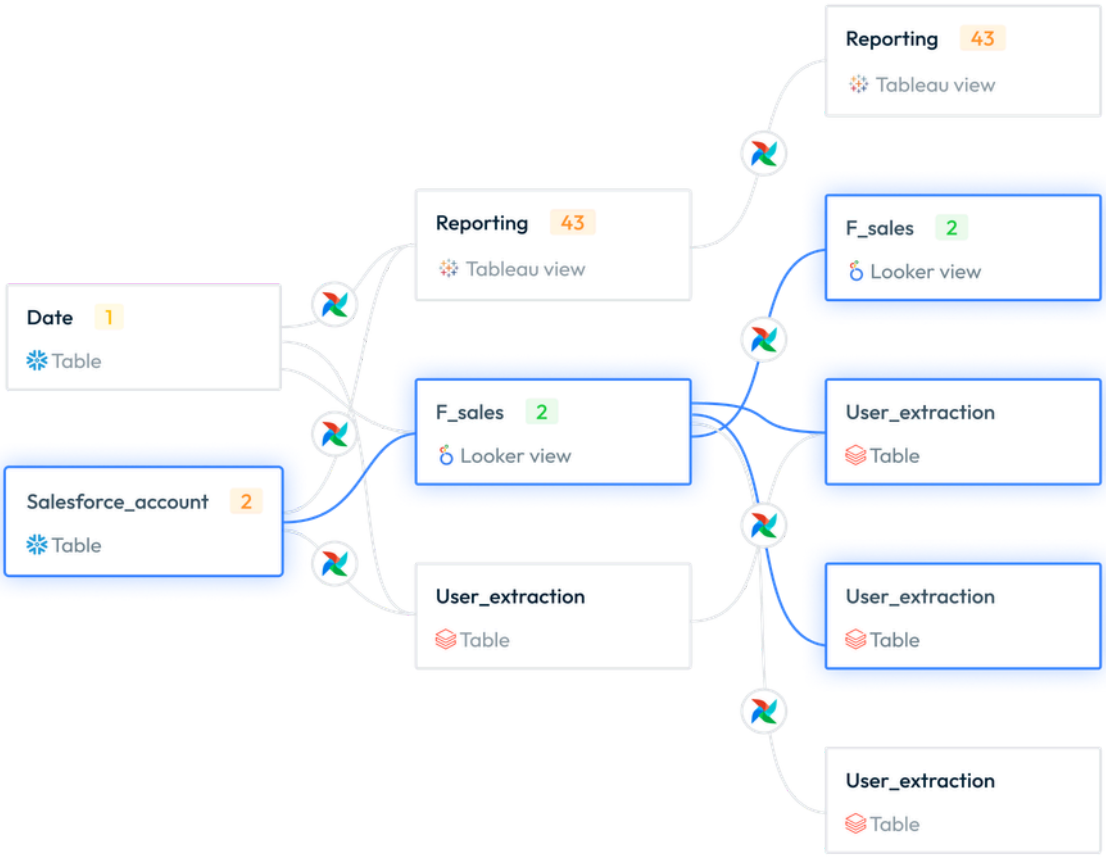
One of the most critical components to efficient root cause analysis and incident resolution is column-level lineage. Column-level lineage (sometimes called “field-level lineage”) maps the dependencies between data sets and tables across data products to understand visually how data moves through your pipelines.

While this is important for understanding your data products at a structural level, it also enables data teams to trace incidents and anomalies back to a source much more efficiently.

Without column-level lineage, data teams will spend hours or even days root-causing a data quality issue before they have the chance to remediate and resolve it. Over the course of a year, that can add up to hundreds of thousands (or even millions) of dollars in lost productivity.

Without column-level lineage, data teams will spend hours or even days root-causing a data quality issue before they have the chance to remediate and resolve it. Over the course of a year, that can add up to hundreds of thousands (or even millions) of dollars in lost productivity.

A good data observability solution will provide this level of dependency mapping right out of the box. Before we go further, let's take a look at an example of how data observability from Monte Carlo might improve root cause analysis and resolution in a real-world scenario.



Root cause analysis is faster with Monte Carlo

Scenario: You're a data leader maintaining a dashboard for revOps and a key metric just shifted by 20%. It could be a real metric shift, or it could be one of a million possible issues with your data stack. What do you do?

Data Quality Issue	Example	Manual RCA	RCA w/ Monte Carlo	est. Time saved per incident
System Level				
Failures	An Airflow job or dbt model fails. Data arrives late.	It's hard to match the job failure alerts from the systems to the impacted tables and dashboards downstream.	You receive a data freshness alert and check the Airflow and dbt sidebars within Monte Carlo. Automated lineage shows the impact downstream.	5 Hours
Modifications	A member of your team makes a change to a Fivetran connector that causes issues downstream.	No jobs are failing so no alerts are sent this time. At some point, you look in the FiveTran logs to see which changes may have impacted the downstream assets in question.	A Monte Carlo alert is sent. You click on the FiveTran sidebar within the platform and immediately see the relevant change.	10 Hours
Permissions & upstream changes	Your orchestrator or ETL system loses access to upstream systems, or a software engineer loads data into a different S3 bucket.	This is one of the more difficult issues to spot. You fire off an inconclusive memo after pouring over code and system configurations.	You get a data freshness alert and quickly triage using insights from our Fivetran or Airflow integrations. Or perhaps you get a query insight for a Snowpipe failure.	5 Hours
Timing	Your dbt model or ETL job runs on schedule, but the source system changes the data delivery cadence.	There are a lot of comorbidities to sift through. You have job failures and queries running against NULLs across your pipeline.	Monte Carlo correlates the freshness issue taking place in the final resulting table to the spike in NULLs in upstream tables. We send you one alert and a coherent story.	10 Hours

Data Quality Issue	Example	Manual RCA	RCA w/ Monte Carlo	est. Time saved per incident
</> Code Level				
Queries	New data isn't loaded when a query runs to update a table. This is a result of a freshness issue upstream.	The logs show a query downstream ran successfully, but it executed on data that never arrived. Even the tried and true "re-run the job" method may not immediately produce results, as the query may not be set to run again for another 24 hours.	Monte Carlo's high correlation insights feature immediately identifies that a data anomaly occurred at the same time that an empty query ran, and points you to the exact issue.	12 Hours
Transform-ation model changes	A member of the analytics engineering team makes a modification to a dbt model that drops columns required for transformation jobs downstream.	Narrowing your investigation to dbt, you might check all recent model modifications. You have some visibility into how each model relates to others, but it is difficult to see which of your tables are impacted. You manually trace all model modifications through your table lineage.	Monte Carlo's dbt and Github integrations link all relevant models to the tables experiencing the anomaly in question. You quickly see all modifications to those models and spot the issue right away.	1 Hour
Upstream Changes	A software engineer updates the code of a microservice, which incidentally changes the schema of the data output.	You check your systems and your code, and spot no recent modifications or issues. You manually follow the data flow upstream before seeing the schema change in the raw data. But you have no context for who is responsible for the data entering this table so you fire off a broad Slack ping and wait.	Data lineage brings you to the most impacted upstream table immediately. Documentation there shows you that Jill in software engineering owns the S3 instance that loads data into this table. You reach out and resolve the issue.	15 Hours

Data Quality Issue	Example	Manual RCA	RCA w/ Monte Carlo	est. Time saved per incident
Data Level				
Third-Party	You recieve data from a third party once a week as part of a contractual relationship. This data, on occassion, is filled with NULLs or incorrect values.	After a few painful emails from stakeholders, you are now aware this third-party data is problematic. You create a calendar notification to remind you to check the incoming data once it's received and compare it to past acceptable datasets.	The data is automatically validated or anomalies are flagged once ingested. SLAs can be established to hold the third-party accountable.	1 Hour
Manual Entry	You recieve data from frontline workers that sometimes has incorrect values.	You manually review the data to ensure values are within a historical norm.	You set an automated field health monitor that alerts you to any distribution anomalies automatically.	1 Hour
Source System Bug	A bug in the source system causes duplicate records, a field to be populated with all NULLs, or some other data issues.	You pull a few ad-hoc queries to validate the records issue and understand the scope, then compute basic stats across time windows to find the “break point.” You must trace upstream dependencies by hand to understand what feeds the table and identify what could cause the NULL or duplicate propagation before searching for the change that caused it in recent PRs, run jobs, deployment logs, etc.	The anomaly is flagged as soon as it appears. You get a clear sign of when the metric deviated and how soon it is, with automated lineage pinpointing the likely origin and guided triage across common failure modes.	8 Hours

Observability agents for root cause analysis

When it comes to delivering AI-ready data, you'll always be limited by your most manual processes. That's why the best observability solutions don't just make your data more reliable – they make your data and AI teams more productive.

Solutions like Monte Carlo's Observability Agents are designed to help teams scale some of the most manual steps of the input reliability equation: namely understanding your data, scaling coverage, and resolving incidents. So far, Monte Carlo has released three Observability Agents for users to leverage in their reliability loops:

- **Monitoring Agent**: Reliability is a team effort. Monitoring Agent helps even non-technical product owners draft and deploy monitors faster.
- **Troubleshooting Agent**: It's not what you can detect that makes the difference. It's what you can resolve quickly. Troubleshooting Agent will investigate anomalies and provide actionable next steps to help data teams resolve incidents 80% faster.
- **Operations Agent**: You don't know what you don't know. Operations Agents will mine insights from your monitor, alert, and asset usage to identify gaps, uncover health trends, and optimize response times.

End-to-end coverage

The other, final, key benefit of a data observability solution is extensibility. Because data observability is a tool designed specifically to monitor and maintain data quality, it's often highly flexible in its application.

Good data observability solutions like Monte Carlo offer integrations across critical platform layers like dbt, storage and compute platforms, and even CI/CD solutions like Github that make it easy to quickly add comprehensive data quality coverage across your entire end-to-end environment.

Choosing the right data observability solution

It's not the most sophisticated solution that drives the most value (though that certainly helps), but it's the solution that you can onboard and operationalize that will actually transform your data quality practice. And agents are fast becoming an important variable in that equation.

If you're in the stage of evaluating potential observability solutions, here are a handful of things to look for in a best-in-class solution:

- Immediate time-to-value
- Product vision
- Enterprise-grade scalability and security
- Automation for baseline monitors
- Immediate time
- AI-powered root-cause analysis
- Agents for critical workflows
- Automated lineage
- Deep ownership controls
- MCP integration
- Unifies data and AI pipelines in a single pane

04 Running with contracts & beyond

Running with contracts & beyond

So you've mastered manual testing and implemented automation through observability to efficiently scale your data quality coverage. Now it's time to turn your attention outward.

As your data platform grows, your stakeholders and domain partners undoubtedly will, too. This creates new challenges in and of itself. So, once you've optimized your team's own data quality practice, the next step is to optimize how your team coordinates cross functionally with stakeholders and partners outside of it.

While increased scale could initiate the run phase of your data quality maturity journey, that's not always the case. In fact, many of the practices in the run phase of your journey are actually fantastic ways to supercharge your data reliability and its usability once you have scalable quality coverage in place. The important thing is nailing data quality coverage and incident resolution first.

So, how do you run?

Running with your data quality motion involves improving both discoverability and accountability – in addition to the explicit quality of your data. And that's accomplished in several important ways.

Developing SLAs

Once you know how to monitor your data – and you have an understanding of what to monitor – you can improve institutional trust in your data products by setting standardized SLAs.

Service-level agreements (SLAs) are a method many companies use to define and measure the level of service a given vendor, product, or internal team will deliver, as well as the solutions if they fail.

In a data reliability context, this gives your data team benchmarks to grade against and tells your expanding stakeholders and downstream data consumers what to reasonably expect from their critical data products.

The more your stakeholders and data products grow – and the more integral those products become to business success – the more essential defining SLAs will become.

Deep monitoring

At scale, you need a baseline level of data quality coverage across your entire platform. However, as you begin to work more closely with your stakeholders and identify which tables and data products are the most critical for downstream consumers, you're likely to identify some tables that require an extra level of attention.

The more important a table is, and the more clearly defined its expectations, the more defined your deep monitoring strategy will become. One example here would be a user-defined monitor that triggers an alert when data fails a specific logic like percent of NULL values or acceptable string patterns.

Data governance

As organizations seek to democratize access to data assets, keeping data safe and standardized becomes a new mountain for data teams to climb. And with access also comes the need for external teams to understand and effectively discover those data assets.

Proper data governance, including metadata management and data cataloging, becomes essential to maintaining and activating the vast amounts of data you're likely to have as your environment grows. In an ideal world, you'll be doing some of this already through dbt documentation, but developing a formalized process and educating your data team and domain leaders will become essential to maintaining governance standards for the long haul.

Data contracts

At this advanced stage, data sprawls across various systems, platforms, and sources. This can result in all kinds of headaches for the data team if not managed correctly.

Let's say, for example, production data from a transactional database lands in the data warehouse and becomes part of a different downstream process. The software engineers in charge of that system aren't aware of any data dependencies outside their system, so when they make an update to their service that results in a schema change, the tightly coupled data systems crash.

Ensuring that teams are aligned in terms of how they use and extract data becomes critical to maintaining the integrity of your data platform and its products. Data contracts are a system of processes and tooling that help data producers and data consumers stay on the same page as your platform evolves.

Data contracts might cover:

- What data is being extracted
- Ingestion type and frequency
- Details of data ownership/ingestion, whether individual or team
- Levels of data access required
- Information relating to security and governance (e.g. anonymization)
- How it impacts any system(s) that ingestion might impact

05 A word on AI-ready data

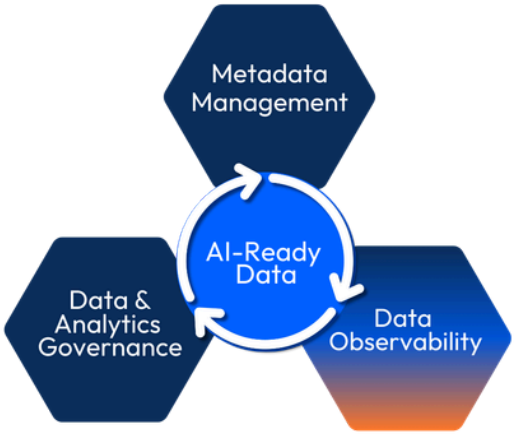
A word on AI-ready data

AI-ready data is still one of the biggest barriers to AI deployments in 2026. You could have the greatest model, the most perfect use-case, the most enthusiastic business users... but at the end of the day, your AI will only ever be as valuable as your data is reliable. And that starts with the quality of your data.

Hear this: getting your data ready for AI isn't easy. But if agents are on your roadmap, then the right mix of tooling, process, and focused energy can get you there.

While the definition of "ready" will vary to some degree based on the use-case (chatbots, agents for financial services, etc.), the concepts are largely universal. If you're preparing your data for production, whether that's agents or otherwise, the end product should look something like this:

- **It's in the cloud**: accessible, discoverable, and able to be quickly activated for a domain-specific use-case.
- **It's being monitored end-to-end**: you have a solution that enables you to observe the entire pipeline for critical quality dimensions.
- **It's certified**: datasets have been validated for quality standards.
- **It has consistent metadata**: standards have been established that dictate how the data will be used.
- **It has documented context**: transparent provenance and lineage is provided for agents to interpret and explain outputs.



Here's the good news: all the work you'll do as you move along the data quality maturity curve is part and parcel to delivering AI-ready data for production use-cases. To put it a different way, governed and high quality data **is** AI-ready data.

Context engineering

Input costs for AI models are roughly 300-400x larger than the outputs. If your context data is shackled with problems like incomplete metadata, unstripped HTML, or empty vector arrays, your team is going to face massive cost overruns while processing at scale.

What's more, confused or incomplete context is also a major AI reliability issue, with ambiguous product names and poor chunking confusing retrievers, while small changes to prompts or models can lead to dramatically different outputs.

Context engineering, which is often baked into an AI-ready data program, is the systematic process of preparing, optimizing, and maintaining context data for AI models. It involves managing metadata and infusing your data with semantic meaning that makes it understandable, both to your team and to the AI models that access it. This can include things like:

- Metric definitions
- Established synonyms (e.g. "sale" = "order" = "purchase")
- Documented relationships
- Registered sample queries to guide agent behavior

In terms of the AI-readiness journey, much of what's considered context engineering would sit firmly in the "run" stage after you've deployed a data observability solution and operationalized a scalable data quality process that prioritizes incident ownership and resolution.

Planning for AI with data observability

> **Gartner expects 60% of the market to adopt data observability by 2026.**
> Gartner Market Guide for Data Observability

The impact of data will only continue to grow as organizations seek to realize the value of AI for the enterprise. But delivering reliable, governable data for non-deterministic use-cases necessitates a system that prioritizes not just reactive data quality monitoring, but proactive incident resolution – with both quick time-to-value and rapid scalability.

Over the last 2 years, data observability has risen to become the preeminent solution to maintain the integrity and reliability of data for AI. This is why Gartner expects 60% of the market to adopt data observability by the end of 2026, driven in part by the rapid adoption of AI-ready data programs.

Because data observability solutions like Monte Carlo provide automated monitoring, lineage, and RCA out-of-the-box, you'll have guaranteed quality coverage no matter how fast you scale. And when AI enters the conversation, integrated solutions like agent observability will give you the ability to quickly unify your data and AI pipelines in a single pane of glass.

If you don't have the right processes and tooling to manage data quality at scale now, you're not going to have the right resources to make your AI reliable later. Of course, traditional practices like manual testing won't disappear completely. Those skills you built in stage one will continue to be useful, but within a more scalable and proactive process. As your data continues to grow and you approach the "run" stage of the Data Maturity Curve (and the inevitability of AI use-cases), you'll be happy to see that you're already set up for a smooth transition.

Go beyond data quality with Monte Carlo

We hope you found this useful. If you did, feel free to subscribe to our YouTube channel, and follow us on LinkedIn where we share new content weekly.

And if you're interested in learning more about Monte Carlo's data + AI observability solution, you know where to find us.

Request a demo