MONTE CARLO

# The Data Quality Maturity Curve

Manual Monitors
Data Observability
SLAs
Data Contracts
GOVERNANCE

# What does your data quality need?

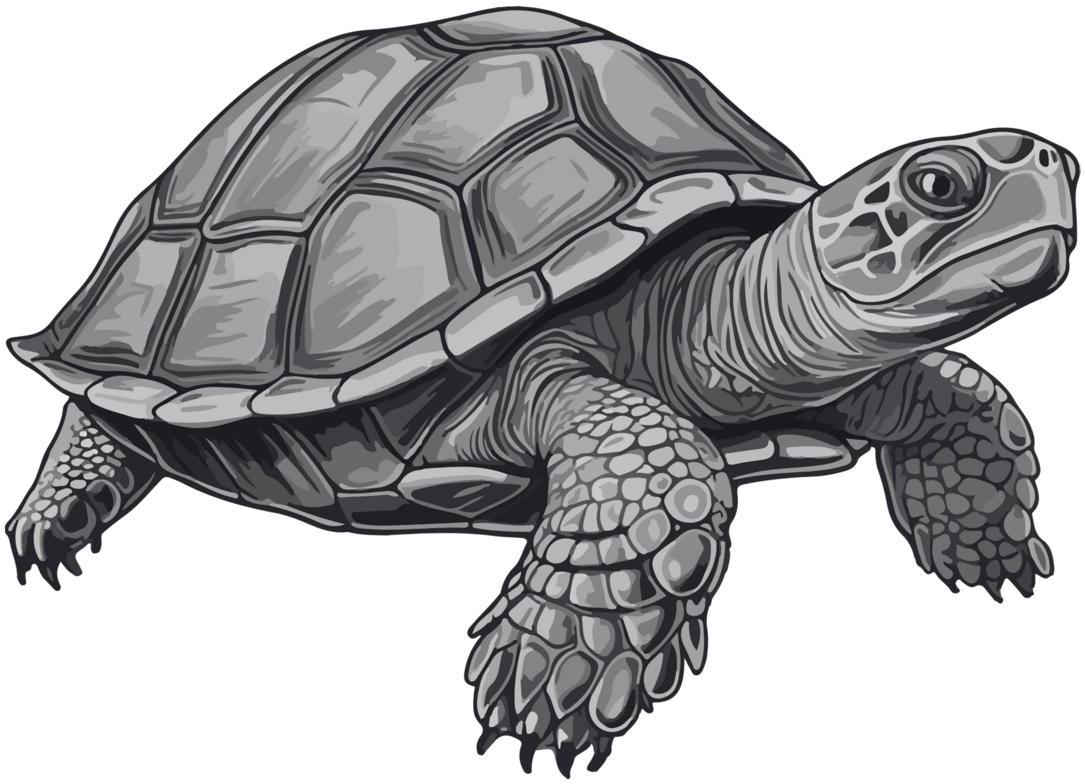
Data quality is important—no doubt about it. Bad data destroys trust in both the data products and the teams that produce them. And it seems like the industry is finally starting to take notice.

But as data quality finally takes its rightful seat at the platform table, where should new data teams start? Great Expectations? dbt tests? Data Observability?

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 works itself out at different stages of your organizational and analytical maturity—to offer some experienced perspective on where you should be at each point in your data quality journey.

Let's get moving.



# Table of Contents

## Introducing 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.**
- **And the transformed data is free from anomalies.**

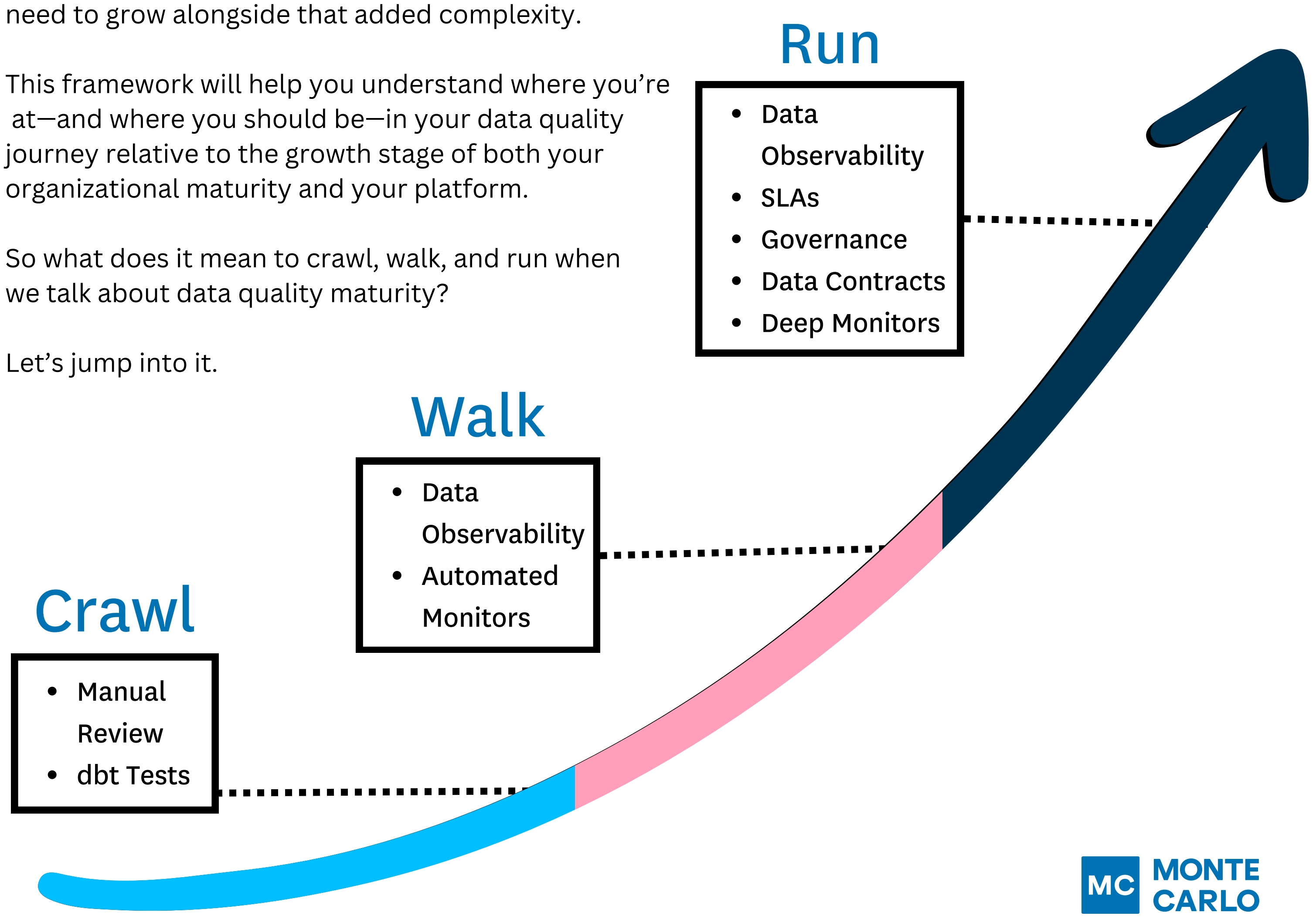
As stakeholders clamor for more and more data, it's easy to secure investments for new pipelines, better discovery tools, and new ways of manipulating and visualizing the data. But all that investment will be for naught if we don't equally invest in the right data quality coverage to deliver the projected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



## 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.

# Crawling with manual monitors and dbt tests

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.

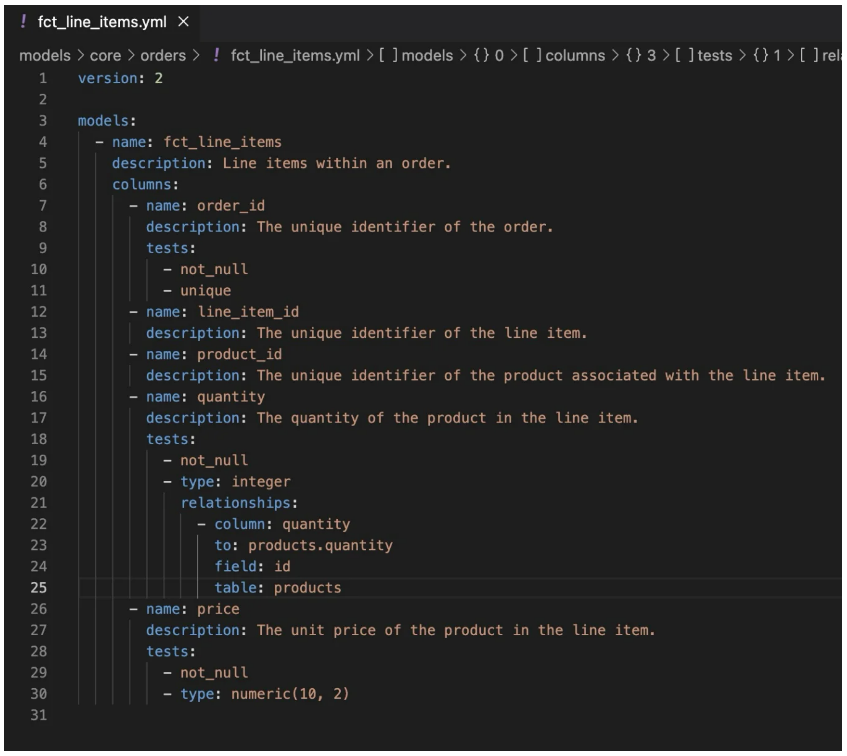
With a limited dataset, data teams can manually inspect tables and columns by simply scanning the data to identify inconsistencies, missing values, and a variety of other data quality issues.

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 pretty quickly. 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 lineage 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.


! fct_line_items.yml ×
models > core > orders > ! fct_line_items.yml > [ ] models > { } 0 > [ ] columns > { } 3 > [ ] tests > { } 1 > [ ] rel

```yaml
version: 2

models:
  - name: fct_line_items
    description: Line items within an order.
    columns:
      - name: order_id
        description: The unique identifier of the order.
        tests:
          - not_null
          - unique
      - name: line_item_id
        description: The unique identifier of the line item.
      - name: product_id
        description: The unique identifier of the product associated with the line item.
      - name: quantity
        description: The quantity of the product in the line item.
        tests:
          - not_null
          - type: integer
            relationships:
              - column: quantity
                to: products.quantity
                field: id
                table: products
      - name: price
        description: The unit price of the product in the line item.
        tests:
          - not_null
          - type: numeric(10, 2)

```

# Walking with data observability

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

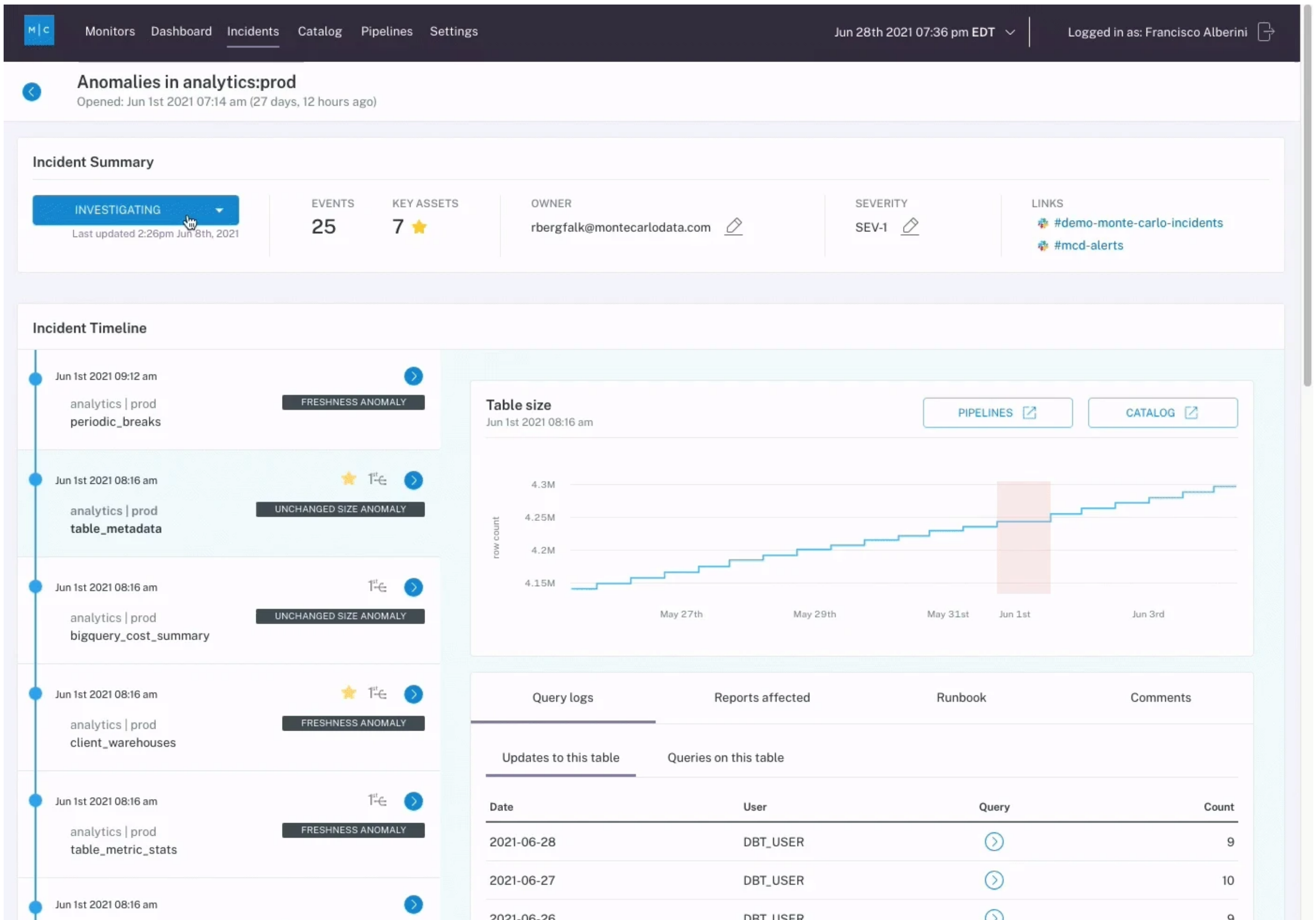
In the same way that walking is the foundation of running, **the second stage of your data quality journey is all about scalability**. And this is where automation comes into play in a big way. As the number of tables and columns grows, and the volume of data increases, manual methods become less efficient. 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 is the priority 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.



Anomalies in analytics:prod

Opened: Jun 1st 2021 07:14 am (27 days, 12 hours ago)

Incident Summary

INVESTIGATING

Last updated 2:26pm Jun 8th, 2021

| EVENTS | KEY ASSETS | OWNER | SEVERITY | LINKS |
|---|---|---|---|---|
| 25 | 7 | rbergfalk@montecarlodata.com | SEV-1 | #demo-monte-carlo-incidents<br>#mcd-alerts |

Incident Timeline

| | | |
|---|---|---|
| Jun 1st 2021 09:12 am | analytics \| prod<br>periodic_breaks | FRESHNESS ANOMALY |
| Jun 1st 2021 08:16 am | analytics \| prod<br>table_metadata | UNCHANGED SIZE ANOMALY |
| Jun 1st 2021 08:16 am | analytics \| prod<br>bigquery_cost_summary | UNCHANGED SIZE ANOMALY |
| Jun 1st 2021 08:16 am | analytics \| prod<br>client_warehouses | FRESHNESS ANOMALY |
| Jun 1st 2021 08:16 am | analytics \| prod<br>table_metric_stats | FRESHNESS ANOMALY |
| Jun 1st 2021 08:16 am | | |

Table size

Jun 1st 2021 08:16 am

| Date | User | Query | Count |
|---|---|---|---|
| 2021-06-28 | DBT_USER | | 9 |
| 2021-06-27 | DBT_USER | | 10 |
| 2021-06-26 | DBT_USER | | 9 |

## 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 as 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.

## End-to-end integrations across your data platform

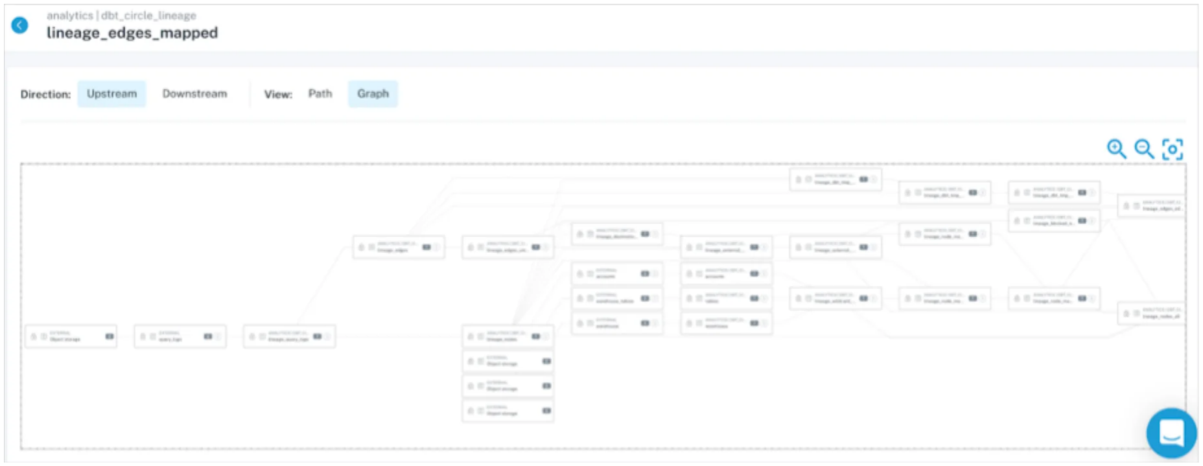
The other key benefit of a data observability solution is extensibility. Because data observability is a tool designed specifically to monitor and maintain data quality, it's often highly flexible in its application.

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.

Looker
snowflake
Google Big Query
Fivetran
dbt
kafka
Apache Airflow
databricks
slack

## 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.

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 metrics just shifted 20%. It could be real or it could be one of a million possible issues with your data or systems. What do you do?

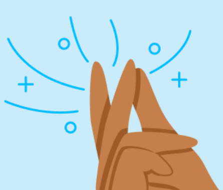
| Example Data Quality Issue | Example | Manual RCA | RCA w/ Monte Carlo | Est. Time Saved Per Incident |
|---|---|---|---|---|
| **System Level** | | | | |
| Failures | An Airflow job or dbt model fails. Data arrives late. | It's hard to match the job failure alerts from systems to the impacted tables 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 finally 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. | Good luck! 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 freshnes 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 |

MC MONTE CARLO

| 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 as a result of a freshness issue upstream. | "Empty queries" are difficult to troubleshoot. 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 a data anomaly occurred at the same time an empty query ran and points you to the exact issue. | 12 Hours |
| Transformation model changes | A member of the analytics engineering team makes a modification to a dbt model that drops columns required for transformation jobs downstream. | Once you have narrowed your investigation to dbt, you might check all recent model modifications. You have some visibility into how each model relates to other models, but it is difficult to see how which of your tables they impact. You manually trace all model modifications through your table lineage. | Monte Carlo's dbt and Github integration links 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. | 12 Hours |
| 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 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. Also schema changes add that | 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 or source system API causes duplicate records, a field to be populated with all NULLs, or some other data issues | Segment sending in wrong data type string instead of integer breaking a lot of stuff, source systems. Automatic change. | FHM and lineage and tagging to the upstream source. | 8 Hours |

## Looking For Efficiencies?

Consider this: The average organization experiences more than 60 data quality issues a month and takes about 15 hours to resolve each issue once detected*.
Can you afford the status quo?



MC MONTE CARLO

*Monte Carlo State of Data Quality Survey of 200 data professionals.

## Planning for the future with Data Observability

While data observability is a relatively recent addition to the data stack, it's quickly becoming one of the most indispensable tools for high performing data teams to maintain the integrity and reliability of their data.

Data will only continue to grow as organizations realize—and capitalize—on the value of their data for business use cases. As you develop your data environment—and by extension your data quality motion—you need to establish a system that provides both fast time-to-value at current scale and enables faster and more reliable scale in the future.

Because data observability solutions like Monte Carlo provide automated monitoring and lineage out of the box, you have guaranteed quality coverage as soon as new data pipelines come online.

Of course, manual testing doesn't completely disappear. Those skills you built in stage one will continue to be useful, but in a more efficient and scalable process. As your data continues to grow and you approach the "Run" stage of the Data Maturity Curve, you will be happy to see that you are already set up for a smooth transition.

## Running with contracts, governance, SLAs and more

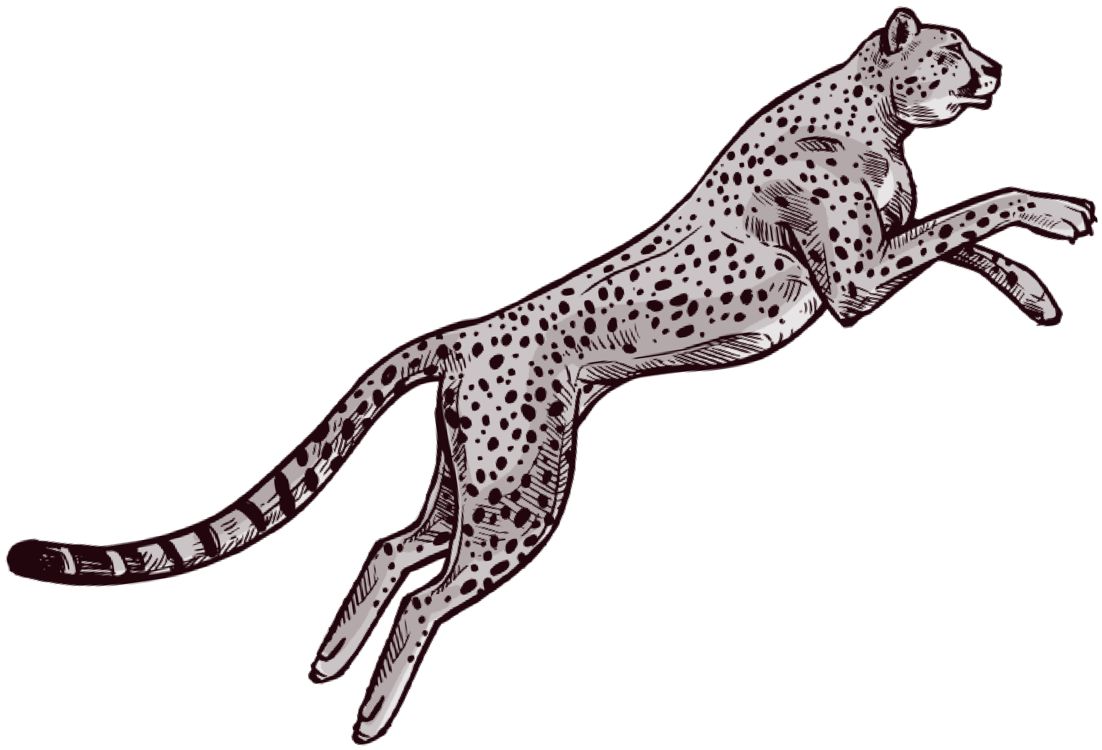
So you've mastered manual testing. You've implemented automation through observability to efficiently scale your data quality coverage. Now it's time to turn your attention outward.

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 your team.

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—and 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 monitoringstrategy 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

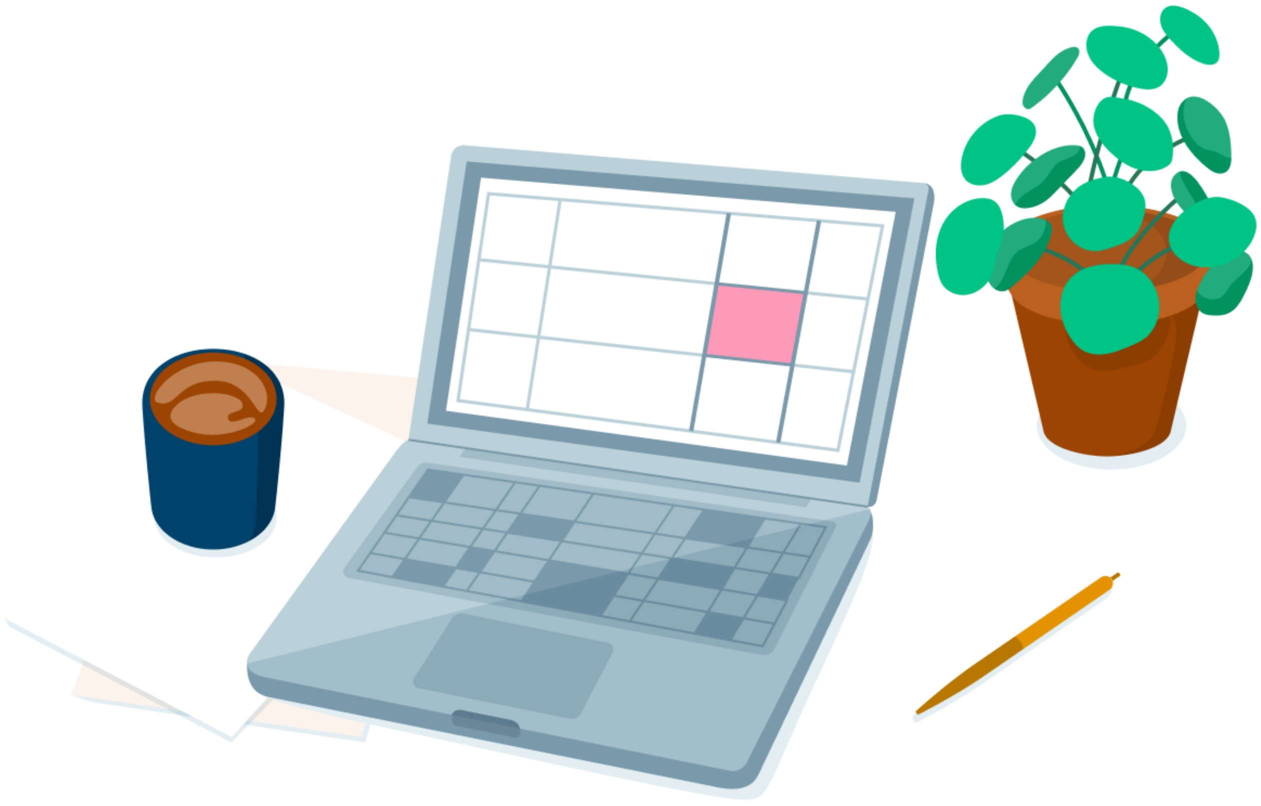
At this advanced stage, data sprawls across various systems, platforms, and sources—and this can result in all kinds of headaches for the data team not managed correctly.

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 process 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



MONTE CARLO

## Build better data products with data observability.

Ready to unlock the real secret to better data products? Request a demo to see how Monte Carlo can help you deliver data products your data consumers can trust.

**Request a demo**

**Looking for more tips & tricks?**
**Check out some of our other resources.**

Data Downtime Blog

Data Quality Fundamentals

Follow us on LinkedIn

Subscribe to our YouTube Channel

**Monte Carlo is the world's**
**#1 Data Observability Platform**

Leader WINTER 2023

Momentum Leader WINTER 2023

Easiest Setup WINTER 2023