### Introduction

Data reliability is a new concept in data operations. At its core, it’s about treating data quality like an engineering problem. This means building reliable systems and processes and implementing practices and tools such as setting SLAs, instrumenting dashboards, monitoring, tracking, alerting, and incident management.

This is in contrast to more ad-hoc solutions for data quality, like SQL checks or having someone whose job it is to debug when the resulting data looks suspect. In this blog post, we will delve into what it means to maintain data reliability across the entire data pipeline, from ingestion to transformation to consumption.

### Data Ingestion

Data reliability in the data ingestion phase means ensuring that the acquired data from various sources is accurate, consistent, and trustworthy as it enters the data pipeline. The goal should be to ingest and process your data only once.

This includes:

* **Understanding your data sources:** Get familiar with the data sources you're ingesting from, which can include:
  + file-based data sources (data lakes)
  + database-based data sources (OLTP and OLAP)
  + SaaS-based data sources (Stripe, Hubspot, Salesforce)
* **Managing your data schemas properly:** Depending on your knowledge of the schema and the frequency of schema changes, your data ingestion pipelines should incorporate technologies that support things like:
  + **Schema evolution:** File formats such as Apache Avro or Apache Parquet enable you to handle schema changes gracefully, allowing for both backward and forward compatibility while minimizing the impact on your existing data processing and analytics workflows.
  + **Schema validation:** Tools like JSON Schema, Apache Avro, and XML Schema Definition (XSD) can be used to define and enforce data structure and data types.
* **Detecting corrupted files and records:** Identify and reject corrupted files or records at the early stage of the data pipeline to ensure high-quality data. For example, if you have a file-based data source, you might check the size of the file. If it’s zero, you can flag that source right at the ingestion stage.

By focusing on these aspects during the data ingestion phase, you can effectively achieve data reliability and lay a strong foundation for the subsequent stages of the data pipeline.

### Data pre-processing and transformation

Depending on whether you have a modern ELT or more legacy ETL setup, the data pre-processing step of the data pipeline might take place in the data warehouse, or in more traditional Spark/Hadoop environments. Regardless of where it takes place and the tools used, the goal of the data transformation step is always to convert from one format or structure to another, often to make it more suitable for analysis, visualization, or integration.

This step is where the bulk of the fine-grained work for producing reliable data occurs. While data reliability during the data ingestion stage is about making sure that the data has arrived at all, in roughly the right structure, during the data transformation stage, the tables are cleaned and checked on a column-level basis. To ensure data reliability in this step, the important steps include:

* **Defining Quality Criteria:** Before you start transforming your data, define the quality criteria for your data, such as accuracy, completeness, validity, consistency, timeliness, and relevance. Align these criteria with your data goals, business rules, and stakeholder requirements.
* **Monitoring data quality:** Once the data has been ingested into the data pipeline, you can begin by setting up checks, either via tests or a more sophisticated data observability solution like Bigeye, that identify data issues like mismatched data types, mixed data values, data outliers, and missing and duplicate data.
* **Cleaning data:** Depending on the type of data you are working with, you may need to address various issues such as:
  + **Missing data:** You can either ignore the affected tuples (if dealing with a large dataset) or manually fill in the missing data (for smaller datasets).
  + **Noisy data:** Techniques such as binning, regression, and clustering can be used to handle noisy data, ensuring that your data is properly grouped and easier to analyze.
  + **Text data:** If working with text data, remove irrelevant elements like URLs, symbols, emojis, HTML tags, boilerplate email text, and unnecessary blank text between words. Verify that text data falls into the appropriate format. Additionally, translate all text into the language of your analysis and eliminate duplicate data.

While data transformation is meant to clean up and improve the reliability of your data, the data transformation step itself can introduce errors or inconsistencies. It’s therefore essential to have guardrails like:

* **Testing transformation logic:** To validate data quality after a transformation, test your transformation logic to ensure that your code or tool is working as intended and producing the expected output.
* **Comparing source and target data:** Compare source and target data to ensure that data values, attributes, and metadata in the target data match the source data or have been transformed according to the specified rules. For example, with Bigeye’s [Deltas](https://docs.bigeye.com/docs/deltas), you can measure the degree of similarity between a source table and a target table and alert when they deviate from each other.

### 3. Dashboarding

We now move to the consumption side of the data reliability pipeline.

Dashboards are a vital tool for visualizing and communicating complex data insights to stakeholders. However, the effectiveness of these dashboards can be significantly impacted by data quality issues. Some tips for implementing data reliability in dashboarding include:

* **Using visual indicators for data quality:** Visually flag data quality issues within the dashboard to alert users to potential problems. For example, you can use color-coding, warning icons, or tooltips to highlight areas of concern. Additionally, consider incorporating data quality scores or metrics in your dashboard to give users a quick assessment of the overall data quality.
* **Incorporating data quality filters:** Allow users to filter dashboard visualizations based on data quality criteria. By enabling users to view only high-quality data, you can ensure that decision-makers focus on the most reliable and accurate information. Additionally, you can provide the option to display data quality issues to help users identify and address underlying problems.
* **Implement drill-down functionality:** Empower users to investigate data quality issues by providing drill-down functionality in your dashboard. (For example, making the SQL query behind a metric on a dashboard easily copy-pastable.) By enabling users to explore the details behind aggregated data, they can gain a deeper understanding of the factors contributing to data quality issues.

### 4. Analytics

The other common use case for data at companies is analytics - as in analysts running SQL queries. Reliable data enables analysts and decision-makers to derive accurate, actionable insights that drive business growth. Best practices for data reliability in the analytics portion of the data pipeline include:

* **Building a data catalog:** A data catalog is a centralized repository that provides a comprehensive view of an organization's data assets. It includes metadata, data definitions, and information about data sources, formats, and relationships. A well-maintained data catalog is essential for data reliability, as it ensures that users have a clear understanding of what data is available, and where it is. This prevents analysts from having to duplicate tables in order to get what they need, which in turn usually leads to discrepancies in the calculations of core metrics.
* **Utilizing data lineage:** Data lineage is the process of tracing data from its origin through various transformations and integrations to its final destination. It helps analysts understand how data is created, modified, and consumed across different systems and processes. By providing analysts with a visual map of your data lineage, they can proactively check to see if their data is being impacted by upstream outages.
* **Setting data pipeline SLAs:** Service Level Agreements (SLAs) are formal commitments between the producers and consumers of data that define the expected performance, availability, and reliability of data pipelines. In the context of analytics, data pipeline SLAs ensure that data is consistently delivered on time and with the required quality for analyst use. For example, a data SLA might be that the data in a certain table must be at most 48 hours old. To establish and enforce data pipeline SLAs:
* Define clear SLA metrics for data pipeline performance, such as data freshness, latency, and error rates.
* Monitor data pipeline performance against SLA metrics and identify any deviations or issues.
* Implement proactive alerts and notifications to inform stakeholders of any potential data pipeline failures or delays.
* Regularly review and update SLA metrics to align with evolving business requirements and data needs.

### Conclusion

Maintaining data reliability along the data pipeline from ingestion to consumption can seem daunting, but there are tools available off-the-shelf that can help. Bigeye is one of them. In the ingestion step, Bigeye can monitor production databases or raw tables as they are loaded into the data warehouse. With Bigeye’s [metadata metrics](https://docs.bigeye.com/docs/t-shaped-monitoring-wide-and-deep), you get out-of-the-box monitoring for freshness and volume, which lets you know that the data pipeline is at least running.

In the transformation step, Bigeye’s [columnar checks](https://docs.bigeye.com/docs/available-metrics) and [Deltas](https://docs.bigeye.com/docs/deltas) feature help you go deeper into your data. They validate that the data not only exists, but is in the right format and cardinality.

Finally, in the dashboarding and analytics step, Bigeye’s [data lineage](https://docs.bigeye.com/docs/lineage) and SLAs allow analysts to find the root cause of their data quality issues.