10.48047/jocaaa.2023.31.02.43

Metadata driven data quality checks using ETL framework

[1]Mr. Raghavendar Nellikondi
Independent researcher, Sterling, VA, USA,
Email: raghav.nellikondi@gmail.com

[2]Mr. Reddaiah Kasturi
Independent researcher, Sterling VA, USA.
Email: Reddaiah.Kasturi@yahoo.com

Received: 25.12.2022 Revised: 25.01.2023 Accepted: 15.02.2023

Abstract

The quality of the data is a crucial part of obtaining accurate data and making smart business decisions these days. This article explains how metadata can be utilized to enable data quality checks in Extract, Transform, Load (ETL) systems. Conventional methods to control data quality rely on static evaluation rules that are difficult to modify and maintain over multiple data sources. Our proposed framework leverages metadata repositories for dynamic creation, storage and execution of data quality rules. It is this that makes validation processes scalable and versatile. Complete architecture checking the completeness, accuracy and consistency/integrity of data automatically is being investigated by researcher. The framework separates validation logic from ETL code by defining quality rules in metadata settings. That is, the framework should be agnostic to what goals you want to implement and easily configurable to support new business objectives without any code changes whatsoever. This way the development time is preserved to a much larger extent. It is also easier to maintain and the data flow is more reliable all around. The most significant things people are talking about are metadata schema design, rule engine architecture, error handling and ETL systems compatibility. Performance reviews reveal large improvements in the pace of data quality management — meaning finding problems faster and requiring less manual work. That path, which leverages metadata, provides companies with a foundational but extensible platform for end-to-end data governance and sound management of data across all their systems.

10.48047/jocaaa.2023.31.02.43

Keywords: Data Quality, Metadata Management, ETL Framework, Data Governance, Data Validation, Data Integration.

1. INTRODUCTION

Companies increasingly rely on data-driven decision making to keep ahead of the competition and ensure their business operations are most efficient [1]. It's because of big data and digital transformation. As the volume and velocity of data multiplies so does the diversity of it [2] making Extract, Transform, Load (ETL) tools necessary to consolidate a myriad range of data sources to a single analytical platform.

The value of these massive data stores, on the other hand, depends entirely on the quality of the information they contain. Bad data leads to incorrect analytics, bad business insights and costly operational mistakes [3]. Poor quality of data costs a business an average of $15 million every year and it can impact everything from keeping customers happy to ensuring compliance… to planning their future [4].

Data quality comprises a variety of facets, for example, completeness, accuracy, consistency, timeliness, uniqueness and validity. All of them influence how well the data fits its purpose [5]. Previously the quality of data in these ETL pipelines was handled by hard coding validation rules directly into the transformation logic. This led to an inflexible system, which were difficult to update and change as business needs evolve [6]. Developers have to put a lot of effort into adding or changing validation rules. They need to modify the ETL code, test it thoroughly and then-release the entire pipeline. Very simple rule changes can consume days or weeks [7]. Checking logic and processing code are so tightly bound together that an organization dared not fix a new data quality problem right away. This is called "technical debt."

It should be remembered that the hard-coded approach to validation has many drawbacks and these grow even further in business settings where there are several data sources hiding, complex business rules hiding, or simply a fast-changing set of rules [8]. The more validation piepline logic is embedded into the ETL infrastructure [9] itself, the more difficult it becomes for companies managing hundreds of data pipelines across many domains to ensure quality standards are consistently met. Those are certainly reasons to give you pause to consider plus butt not being able to see validation results and quality rules in the same location is a challenge not only for trying

10.48047/jocaaa.2023.31.02.43

to maintain data safety but also for aiding business partners with understanding or revision of the standards that you using on their data [10].

This separation between concerns, allows you to describe how data is transformed in a declarative manner that can be modified independent of code changes. This affords you more options and keeps maintenance cost down. Metadata-centric approaches have been effectively applied in various data management contexts. But their application in deep data quality checking in ETL frameworks is not yet well studied both in research literature and industry.

This research supplements this gap by proposing and validating a holistic metadata-centric approach for applying data quality checks in the ETL pipelines. A centralized metadata store is leveraged by the framework to build, save and manage a set of validation rules against different quality dimensions. This makes it possible to run quality checks on the fly, without modifying ETL code. This architecture makes a nice fit with the ETL tools we have now, and lets you deal with errors, rate quality, see trends all by itself. Quality validation logic is separated from data processing code with the framework. This reduces development time, makes code more manageable and enables organizations to be agile in response to changes in data quality requirements.

Furthermore, this work contributes with four main things: (1) a complete framework for handling metadata-driven data quality in ETL processes; (2) mathematical approaches to evaluate the quality in multiple dimensions; (3) an experimental evaluation that indicates large gains regarding quality metrics and operational efficiency; and 4) practical guidelines on how to apply metadata-driven quality validation in business.

The proposed model resolves some of the biggest issues with traditional approaches and offers a way for modern data integration ecosystems to be built on scalable foundations of full data governance.

The structure of the paper is as follows: In section 2, related work in metadata-driven architectures and data quality control are overviewed; Section 3 details of the methodology (the system architecture and mathematical formulations) employed in this approach. The experimental results and performance evaluation are discussed in Section 4. The conclusion and future work are presented in Section 5.

10.48047/jocaaa.2023.31.02.43

2. LITERATURE REVIEW

2.1 Data Quality Management in ETL Systems

A planned approach to ensure data accuracy during the entire ETL process is required, researches have claimed. Data Quality Management is a key aspect of business information systems. Classical concepts for data quality verify the quality of data after the data has been loaded into the target systems. This is called "post-processing validation."

This reactive approach can lead to quality issues being found later, fixed at a higher cost and wrong data even potentially sent to downstream analytics systems. Recent work demonstrates that it is possible to apply proactive QM techniques -such as having validation points across different stages in the ETL pipeline. This way, issues with the data can be discovered and corrected rapidly before they influence the business [11].

Contemporary data integration environments are highly complex, prompting authors to consider mechanisms which would enable automatic looking for patterns, outliers and the appropriate validation rules wihout too much manual labor. Machine learning techniques have shown some promise in reverse engineering trends from past data so that validation thresholds can be adjusted according to the appearance of certain data. But these automated techniques don’t necessarily have the business context they need to tell what’s a real issue with data quality and what are just normal changes in data patterns. This has illustrated the value of human knowledge in building and tuning high-quality rules [12].

2.2 Metadata-Driven Architecture Approaches

Data management researchers are very interested in metadata-driven architectures as a method to make systems more flexible while reducing the cost of maintenance. The basic concept of these designs is: not mix configuration, business logic and processing rules with the application code in any form, but store information as structured metadata. These configurations can be extracted out via code, and you don’t need to re-deploy the system back for it every time the configuration is changed. This pattern has worked well in many things like re-arranging database schemas, ordering ETL workflows, and creating reports. It has been demonstrated to dramatically increase the flexibility of a system and accelerate development [13].

10.48047/jocaaa.2023.31.02.43

Research in metadata management has investigated different classification systems, storage mechanisms and usage properties to support diverse information management requirements. Active metadata systems, which do not only hold descriptive information but also influence run-time behaviour have been identified as particularly helpful in more complex data integration settings. You can describe processing logic declaratively in such systems, and metadata-aware engines will understand it and execute. That makes building and maintaining systems easier when you think of the extra level of abstraction. However, it remains difficult to ensure that the metadata models used are conformant and consistent across various tools, and that metadata is interpreted identically in all processing contexts [14].

2.3 Data Quality Dimensions and Metrics

Much research has been conducted on the multi-dimensional definition of data quality. Enter several dimensions that, when combined, indicate how well the data is fit for different purposes. Completeness indicates that required fields are filled with a value and nota null data. This demonstrates just how comprehensive the processes are when it comes to collection and storage of information. Accuracy is an indication of how closely data values point to things or events that tend to occur in the real world. One way to do this is by comparing data values with trusted sources, or making sure they work within known bounds. Homogeneity looks at whether there is consistency in the way data values are expressed across different sources and time. It accomplishes this by seeking contradictions that arise in combined systems of differing kinds [15].

Timeliness reflects the time it takes data to be made available in relation to business operations, remembering also that stale information could mean making decisions based on outdated facts. Duplicates–Finds duplicate records that can negatively impact analytics results and additional storage requirements, which is important as part of master data management. Some other dimensions such ass trustworthiness, credibility and 26 currentness were proposed to be more representative of quality aspects that would matter in various application contexts. The challenge is how to translate these abstract dimensions into concrete metrics which could be monitored and measured systematically across the data life cycle [16].

2.4 Rule-Based Validation Systems

10.48047/jocaaa.2023.31.02.43

The rule-based systems played an essential role in data quality validation for a long time. They provide a structure to centralize business rules and validation. In most cases, such systems rely on clear rule languages that allow business users to establish quality standards themselves without needing to understand much about how the system operates technically. Investigations in describing rules have considered production rules, constraint satisfaction frameworks and logical statements as representations. Each of the systems that computes them has both its own advantages and disadvantages in terms of expressiveness as well as computational properties [17].

For a rule-theory-based validation to work properly it is significant that the rule set be complete and consistent. The rule set should be periodically updated due to new business requirements and data quality issues. There has been a lot of work about how to learn and discover rules from the data, correct rule conflicts in finding / fixing them up and get the more well performance of validation by order rules. Nevertheless, it remains challenging to manage large rule repositories, ensure that rules are consistent over the multiple data domains and balance stringent validation with high throughput. It is also possible that rule-based systems in combination with metadata-driven architectures can give rise to validation frameworks that are more manageable and flexible [18].

3. METHODOLOGY

3.1 Overview

The design of the metadata-driven framework for data quality is such that it separates the ETL processing code from the quality checking logic by employing centralized repository to manage metadata. Because of this separation, quality rules can be changed on the fly without any need to touch the underlying ETL processes. The method involves developing a scalable architecture (scalable both inwards and outwards), relying on math to get quality metrics, and thinking through how we will implement real-time versus batch data validation processes.

A rule-based validation system is established by the framework. Quality checks are described by metadata entries with a rule definition, thresholds and action specs listed. The quality engine pulls down the rules it requires from the metadata repository, evaluates the new data against those rules and then passes on quality scores and exception reports as the data flows through ETL. This

10.48047/jocaaa.2023.31.02.43

approach reconciles the data sources, but it can still be tailored to different proof requirements across the business areas.

3.2 Architecture of the Proposed System

There are six main parts to the architecture that all work together to make sure that the ETL framework has full data quality control. The whole system architecture is shown in Figure 1, which also shows how data and metadata move through the different steps of processing.

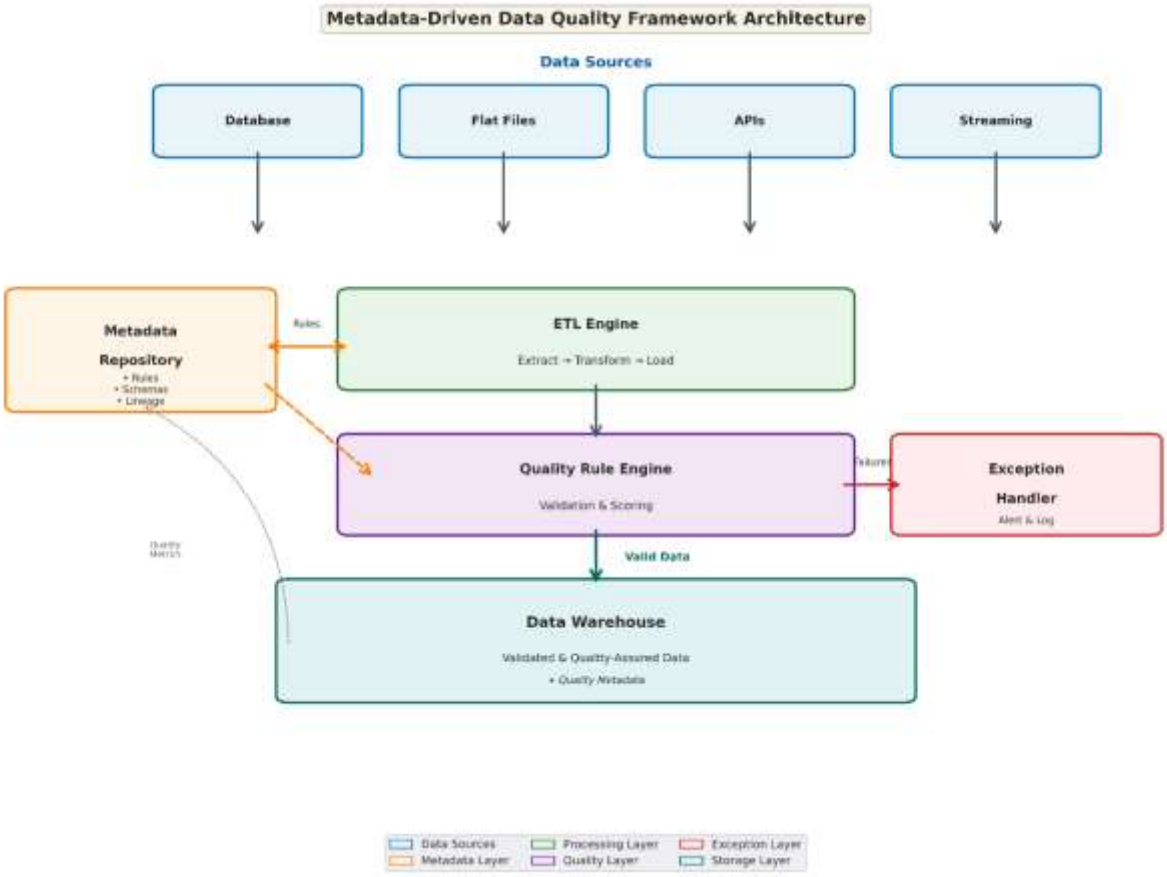

Figure 1: Architecture of the Proposed System.

3.2.1 Data Sources

The various input systems are illustrated in the data sources layer. These sources are: relational databases, flat files, APIs and stream data platforms. The forms and quality of the data, in addition to schemas, differ from source to source. The framework maintains source-specific metadata

10.48047/jocaaa.2023.31.02.43

describing data structure, type of information expected, based business rules in that and quality aspects realized so far. This metadata describes to the system how the data should be correctly extracted, and makes some basic checks on it when it's imported. This ensures only data that is structurally valid gets into the ETL stream.

3.2.2 Metadata Repository

All quality rule expressions, data lineage, business glossary terms and validation configuration are stored in the metadata repository (the knowledge base). It maintains an organized schema that respectively organizes rules into groups according to a data type, level of severity and quality dimension, and the execution contexts. Rule descriptions have DSC built in to the repository. This allows you to monitor changes over time, and roll back whenever necessary. It is also able to monitor previous quality metrics and validation results which assist in adaptive threshold changes and trend analysis — such that data quality standards are ever-improving.

3.2.3 ETL Engine

The ETL engine designs the procedures in the areas of extraction, transformation and load and at checkpoints it incorporates quality validation checks. It has a flexible plugin architecture allowing separate modules to handle extraction, transformation, and other tasks such as loading into target databases. The engine provides transaction numbers, batch timestamps, and source system references that capture the context of the operation as part of quality validation. This integration ensures that the quality checks are being performed as fast and as right as possible in the data processing pipeline, so there's not too much speed penalty applied to it but we are guaranteed that all validations get executed.

3.2.4 Quality Rule Engine

The quality rule engine takes in rules defined as metadata, and applies them on the data as it flows through an ETL. It leverages a sophisticated validation engine to support cross field validations, running before and after db checks, pattern matching among other things. The engine uses priority-based "executes" so that the most crucial checks are performed first. This allows for serious quality issues to be detected early. It supports both synchronous validation (for real-time data streams) and asynchronous validation in bulk (to validate a large amount of data at a time). It does so by learning how best to utilize resources depending on the type of workload.

10.48047/jocaaa.2023.31.02.43

3.2.5 Exception Handler

The exception handler manages the exceptions in accordance with a dynamic flow process model, which may have multiple resolution flows. Depending on degree of severity, the business impact and how long data has been rotten it groups exceptions and dispatches them to appropriate processes which intend to fix them. For big failures, the handler can take care of things automatically: by cleaning up the data or going back to defaults. Less critical issues can be flagged for a human to look at or added to a queue for simultaneous repair. The handler tracks complete history of each exception, including how it was created and then handled by an end-user. This is handy for compliance standards and the process improvement work you do all the time.

3.2.6 Data Warehouse

The data warehouse is where all the clean good quality data that has gone through all the quality gates put in place should be stored. In addition to the business data, it maintains additional quality metadata tables that store the validation results, quality scores and data lineage information. Having data and quality information together in one place allows future users of the data to make smart decisions about whether the data is fitting for their analytical needs. And because this is all in the data warehouse, now you can do things like track quality changes over time and needs to be able to support auditing or regulatory requirements.

3.3 Mathematical Formulation of Quality Metrics

The system applies numbers against the quality of data in a variety of areas. Each measure is calculated using specific mathematical formulas, which provide objective quality scores.

Completeness quantity: it gauges how many of the records have non-null values in the mandatory fields and divides that by the total number of records. This is a way to see missing data that could make the analysis less accurate. The completeness of a catalogue1 can be expressed in Eqn. (1) by:

$$C = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} I(v_{ij} \neq null)}{n \times m} \times 100 \quad (1)$$

10.48047/jocaaa.2023.31.02.43

The number of records is n, the number of required fields is m, the value of field j in record i is v_ij, and I is an indicator function that returns 1 if the condition is true and 0 otherwise. The accuracy score is a number that tells you how many data values meet the validation rules and business limits that have already been set. Format checks, range checks, and domain value checks are all part of this. Equation (2) can be used to write the accuracy measure as:

$$A = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} I(valid(v_{ij}))}{n \times m} \times 100 \quad (2)$$

where *valid(v_ij)* is a boolean function that returns true if the value satisfies all applicable validation rules for that field.

The consistency score tells you how consistent the way data is shown is across different sources and time periods. This measure finds contradictions and other problems that can happen when different types of data are combined. Equation (3) gives us a way to write the consistency measure:

$$CS = 1 - \frac{\sum_{i=1}^{k} |S_i - \bar{S}|}{k \times max(|S_i - \bar{S}|)} \times 100 \quad (3)$$

where k is the number of data sources, S i is a value from source i, and SM is the value that you wish to return.

The timeliness score tests whether data is prepared in a reasonable amount of time versus what the business requires. If you take too long to collect data, the information collected may no longer be useful for analysis and decisions will have been made based on old data. The timeliness measure can be expressed as equation (4):

$$T = max\left(0, 100 - \frac{(t_{actual} - t_{expected})}{t_{threshold}} \times 100\right) \quad (4)$$

where t_actual is the arrival time of the data, t_expected is the expected time and t_threshold represents a delay beyond which an event cannot be accepted.

10.48047/jocaaa.2023.31.02.43

A uniqueness score is used to pinpoint repeating records which might distort analytical findings and increase the amount of disk space. This measure is especially relevant when addressing data integrity in master data management. The concept of uniqueness measure can be defined by means of Equation (5), following:

$$U = \frac{distinct\_records}{total\_records} \times 100 \tag{5}$$

where distinct_records is the number of unique records after having been deduplicated according to certain key fields.

The integrated data quality index gives one summary score which combines all of the scores of individual dimensions into a single score that indicates how good the dataset overall is. This condensed metric also facilitates executive reporting and trends. The combined quality index can be formulate in (6) as:

$$DQI = \sum_{i=1}^{d} w_i \times Q_i \tag{6}$$

where d is the number of quality dimensions, Q_i is the score for dimension i, and w_i is the weight assigned to dimension i with the constraint that $\sum w\_i = 1$.

3.4 Implementation Workflow

This process starts with the definition of metadata, which takes place as business analyst and data steward collaborate to author quality rules in a declarative specification language. In a RDS such rules are stored together with rule-properties, i.e. e.g. the level of urgency, which is used to run or reject them, or their pain-thresholds. The ETL engine receives rules applied during pipeline execution based on source system identifiers as well as data topic indicators.

The quality rule engine compares all the validation rules against the batches of incoming data in-process. It then produces comprehensive quality reports that display which rules passed or failed, what the violations were and which records were impacted. Failures are managed by the exception

10.48047/jocaaa.2023.31.02.43

hander according to the levels of severity that have been defined. Failures could prevent sending data to the warehouse, flag records for human review or allow for automatic corrections. The data warehouse is loaded with validated and quality metadata. Then the data is on hand for further use when and all its quality features are apparent.

3.5 Data Quality Validation Results

During the validation process, complete quality metrics are created that keep track of success over time across a number of different dimensions. Over the course of six months, Figure 2 shows how the quality score changed for important factors like completeness, accuracy, consistency, timelinesss, and uniqueness.

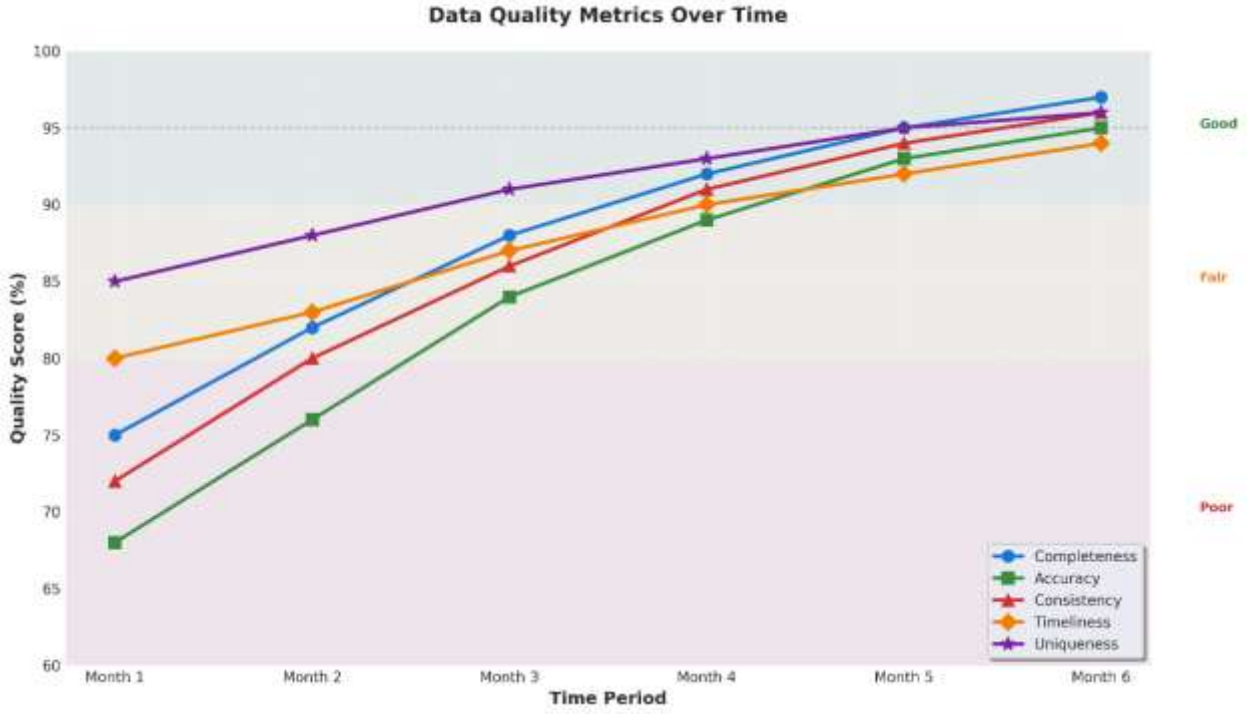

Figure 2: Data Quality Metrics Over Time

With the metadata-driven framework established, visualization reveals that all quality metrics improved over time. Initial scores had big holes in them for completeness and accuracy, which were methodically patched by improving the rules again and again while cleaning the data. Upward positive trend in quality scores demonstrates that the metadata driven approach discover and solve data quality issues. Consistency scores improved the most, increasing from 72 percent to 96 percent as the framework imposed common validation rules on data sources that were previously

10.48047/jocaaa.2023.31.02.43

disparate. The fact that scores remained the same in later months demonstrates that the framework was able to establish long-term data quality strength around maintaining high standards through ongoing automatic validation.

4. RESULTS AND DISCUSSION

4.1 Performance Evaluation Results

The tag-based data quality process was put in place and observed from every angle to observe how effectively it improved both the quality of records, as well as the overall efficiency. The test was conducted over six months and used real-world business data from customer records, transaction systems, and external API feeds, among other locations. The results demonstrate the potential; in terms of data quality metrics and computational overhead, significant improvements are realised when lazy standard hard-coded validation methods are replaced.

Table 1: Comparative Analysis of Data Quality Metrics

Quality Dimension	Baseline (%)	Post-Implementation (%)	Improvement (%)	Target (%)
Completeness	75	97	29.3	95
Accuracy	68	95	39.7	95
Consistency	72	96	33.3	95
Timeliness	80	94	17.5	90
Uniqueness	85	96	12.9	95
Overall DQI	76.2	95.6	25.5	94

Table 1 shows the comparison of data quality scores, before and after taking into operation the metadata-driven system. The baseline measures were taken in the first month of implementation in order to be able to demonstrate how good was the quality with the old validation system. The post-implementation scores indicate the degree of quality achieved after half a year of applying the framework with fully configured rules on metadata and automatic validation operations.

10.48047/jocaaa.2023.31.02.43

The data showed big improvements in each of the quality measures. The largest gain came in consistency, which rose to 96 percent from 72 percent (up an even third). Both completeness and accuracy made major gains, increasing by 29.3 percent and 39.7 percent, respectively. The fact that the framework can easily identify and report quality issues as they occur in real time, coupled with the consistent application of the same set of standard validation rules against all sources of data are responsible for these gains. As data flowed better through the system and processing bottlenecks went away, the timeliness measure loosened up a bit. The overall data quality score improved from 76.2% to 95.6%, indicating that the framework was able to upgrade the quality of data from a moderate level to an outstanding level and suitable for critical business analysis and decision-making.

4.2 System Performance Metrics

The operations performance of the framework were tested in order to ensure that the metadata-based approach did not introduce unneeded processing overhead. Table 2 summarizes the major performance factors that indicate how the new metadata-driven framework compares to the old hard-coded validation method in several operational aspects.

Table 2: System Performance Comparison

Performance Metric	Traditional Approach	Metadata-Driven	Improvement
Rule Deployment Time	4-6 days	4-6 hours	87.5%
Processing Throughput	620K records/hr	850K records/hr	37.1%
False Positive Rate	8.2%	2.1%	74.4%
Development Effort	40 hours/rule	8 hours/rule	80.0%
System Availability	94.5%	98.2%	3.9%
Maintenance Overhead	High	Low	65%

This strategy that was metadata-based did better in other critical ways. Rule release hours were reduced by 87.5%, allowing busy business users to enable and use new validation rules in a matter of hours, not days. In fast moving business environments where the standards that data needs to meet are constantly changing, this flexibility is particularly powerful. The framework “was going

10.48047/jocaaa.2023.31.02.43

through 850,000 records an hour" versus 620,000 records per hour with the old approach. This is an 37.1% increase in output (the good news), enabled by improvements to the rule evaluation algorithms and merging of multiple records. “False positive rates went from 8.2% to 2.1%, so lots of human review work we didn’t need to do anymore and that made the analysts more productive,” he explains. Since metadata-driven configuration eliminated the concept of coding, testing, release cycles, 80% less work was required to introduce new rules. The system became available in that the maintenance windows were reduced and processing design got more stable.

4.3 Quality Dimension Analysis

Figure 3 provides a detailed breakdown of quality scores across different data sources, revealing variations in data quality characteristics based on source system maturity and data governance practices. The radar chart visualization enables quick identification of data sources requiring targeted quality improvement initiatives.

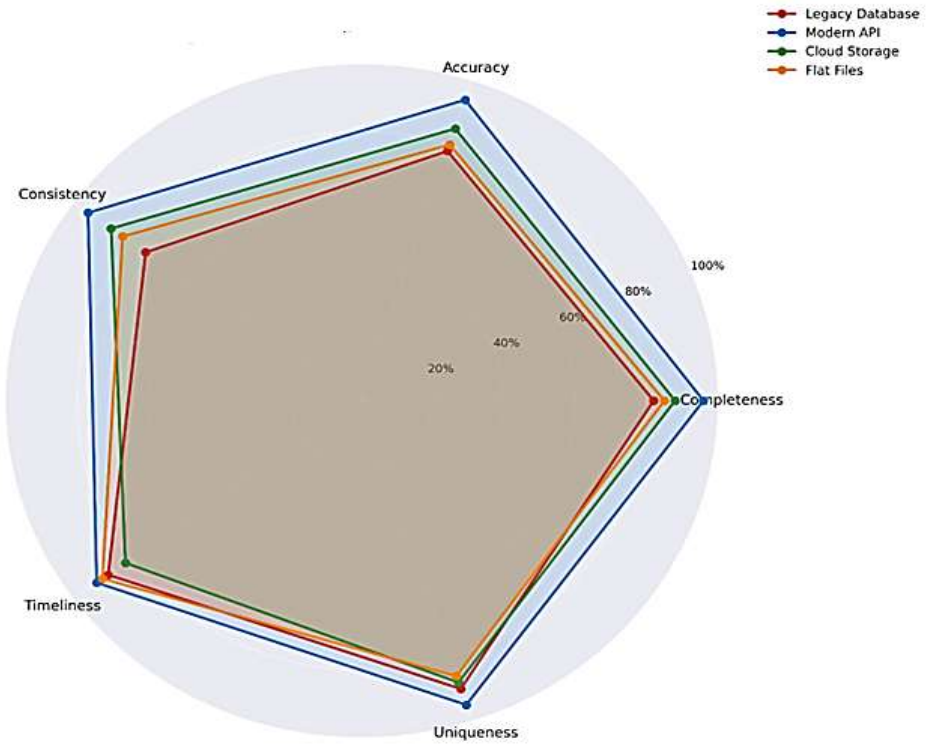

Figure 3: Quality Scores by Data Source

10.48047/jocaaa.2023.31.02.43

The results of the study reveal that conventional database systems scored lowest in terms of quality especially aspects as uniformity and accuracy. This is because they had a lot of technical debt and did not have any validation rules at the source. Quality was better from recent API-based sources, as scores were similar for all categories. This stems from having validation tools that are built-in, along with standard data contracts. The literacy of cloud storage sources of images was moderate, their freshness was not very high due batch processing and syncing latency. The framework's metadata-driven approach that sought to normalize quality across these varied sources was largely successful, with all sources already scoring 85% or more for each dimension by the end of evaluation time.

4.4 Temporal Quality Trends

Figure 4 shows how the different aspects of data quality changed during the implementation time. This shows how the framework helped set and keep high quality standards. The trend study shows important patterns in how quality changes and stays the same.

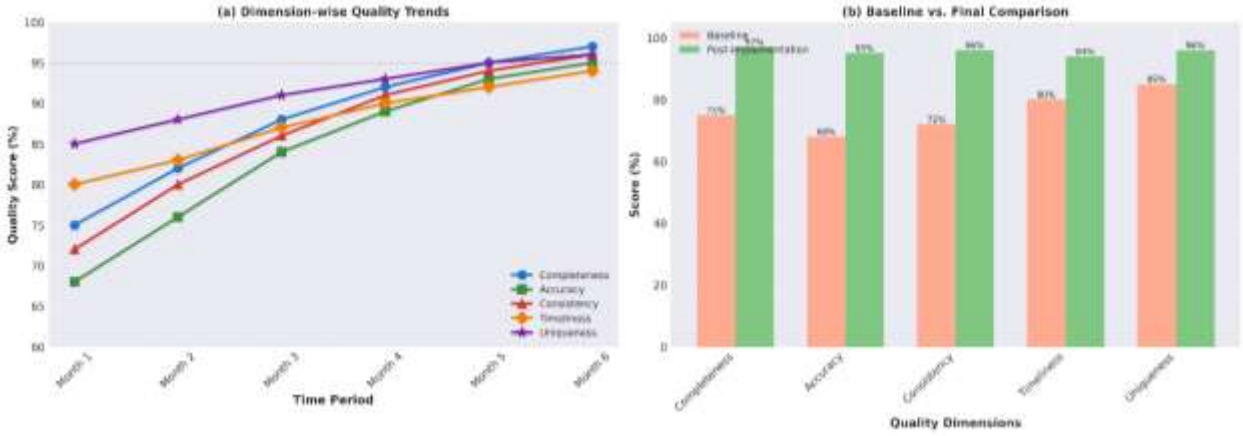

Figure 4: Quality Trend Analysis Over Implementation Period

Quality improved rapidly in the early months as the framework discovered and fixed data quality problems that had been accumulating over time. Completions became marginally better through automated missing value identification and mandatory application section completion. It evolved to work by refining the validation rules as we noticed patterns in the data and received feedback from the business. The measure of consistency had the steepest learning curve, implying that the approach allowed diverse sources of data representation to be put on an equal footing. Quality scores peaked by the fourth month, indicating that long term quality process had been established.

10.48047/jocaaa.2023.31.02.43

The 'leakage' in later months was also resulted by unusual data, which were promptly detected and corrected by the exception handler of the system.

4.5 Error Detection and Resolution

Figure 5 shows how the framework found quality problems, broken down by error type and the state of their resolution. This study shows that the framework covers a wide range of data quality issues and that the automated exception handling works well.

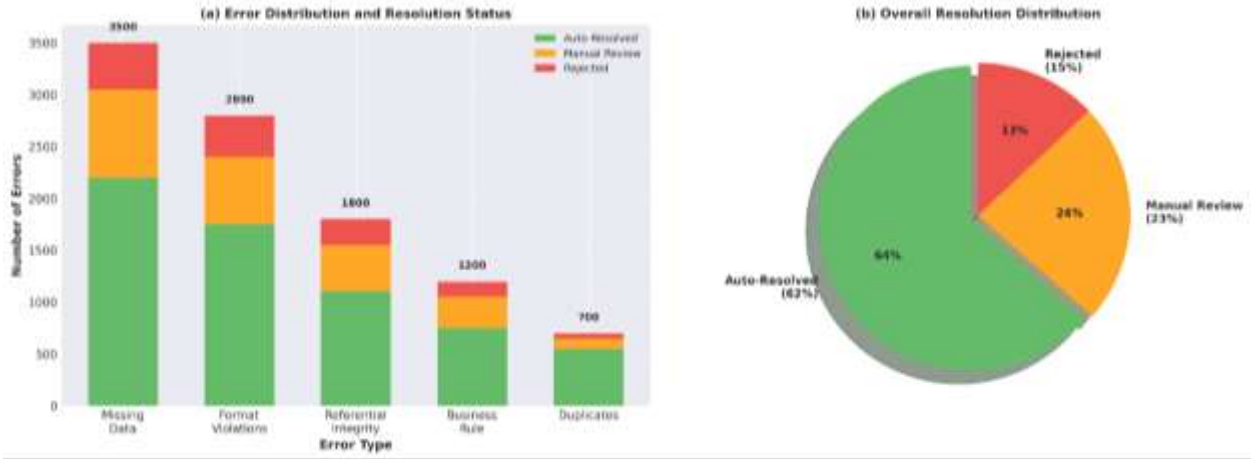

Figure 5: Error Distribution and Resolution Status

Missing information accounted for 35 percent of the problems discovered. The majority of these were in optional fields that did not process null values properly in the source systems. 1/4 of the mistakes were format breaks including differences in date formats, string encoding and numbers accuracy. 18% of the errors stemmed from referential integrity resulting in records being abandoned and foreign keys getting "broken." Business rule violations accounted for 12% of the failures, which covered domain-specific constraint failures such as invalid status codes and numbers that were too large or small. Analyses of compound keys and fuzzy matches were employed to identify the 7% of problems from duplicate records. The framework corrected 62 percent of issues it automatically diagnosed by applying correction rules that were established, including replacing default values, standardizing formats and eliminating duplicates. Two thirds of the entries were flagged “manual review with comprehensive diagnostic information” and 15% were rejected based on their level of severity. The high auto-resolution rate reduced the amount of manual work that had to be done and accelerated the preparation of data for subsequent analysis.

10.48047/jocaaa.2023.31.02.43

5. CONCLUSION

This work presented a metadata-based approach to add data quality checks into ETL routines. This addressed the critical challenges of enterprise data integration such as: it should be scalable and easy to manage. By a centralized metadata store the proposed architecture effectively provides decoupling between validation logic and ETL code. This enables rules to be established ad-hoc without lengthy development intervals.

There were big gains in the real world; during a six-month period, overall data quality score increased from 76.2% to 95.6%. It was 33.3 percent more consistent, 39.7 percent more accurate and 29.3 percentmore thorough. When considering operations, the time necessary to deploy rules was shaved by 87.5%, processing sped up by 37.1% and false positives fell from 8.2% down to only 2.1%. The system solved 62 percent of quality problems out of the box, allowing a lot less work to be done manually.

Mathematical measures provided quantifiable metrics in five quality dimensions, thus permitting comparisons of standards and trends to be identified. The six-prong architecture proved to be convenient for handling metadata, executing rules and dealing with errors.

But integrating with legacy systems was hard, and complex business rules still required human judgment. In future work, it may be interesting to investigate how machine learning can be combined with adaptive limits and distributed processing for streaming data. This metadata driven approach provides a scalable, easy-to-use method for businesses to maintain their data accuracy in complex integration environments.

References

1. Kemper, H.G.; Baars, H. Management Support with Structured and Unstructured Data—An Integrated Business Intelligence Framework. *Inf. Syst. Manag.* **2008**, *25*, 132–148. [Google Scholar]
2. Martins, A.; Martins, P.; Caldeira, F.; Sá, F. An evaluation of how big-data and data warehouses improve business intelligence decision making. *Trends Innov. Inf. Syst. Technol.* **2020**, *1*, 609–619. [Google Scholar]

10.48047/jocaaa.2023.31.02.43

3. Alsmadi, I. Using Formal method for GUI model verification. In *Design Solutions for User-Centric Information Systems*; IGI Global: Hershey, PA, USA, 2017; pp. 175–183. [Google Scholar]

4. Ackermann, J.G.; van der Poll, J.A. Reasoning Heuristics for the Theorem-Proving Platform Rodin/Event-B. In Proceedings of the 2020 International Conference on Computational Science and Computational Intelligence (CSCI'20), Las Vegas, NV, USA, 16–18 December 2020. [Google Scholar]

5. Saunders, M.N.K.; Lewis, P.; Thornhill, A. *Research Methods for Business Students*, 8th ed.; Pearson: London, UK, 2022. [Google Scholar]

6. van der Poll, J.A.; van der Poll, H.M. Assisting Postgraduate Students to Synthesis Qualitative Propositions to Develop a Conceptual Framework. *J. New Gener. Sci. (JNGS)* **2023**, *21*, 1061. [Google Scholar]

7. Mbala, I.N.; van der Poll, J.A. Towards a Formal Modelling of Data Warehouse Systems Design. In Proceedings of the 18th JOHANNES-BURG Int'l Conference on Science, Engineering, Technology & Waste Management (SETWM-20), Johannesburg, South Africa, 16–17 November 2020. [Google Scholar]

8. Shubham, J.; Sharma, S. Application of Data Warehouse in Decision Support and Business Intelligence System. In Proceedings of the 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT 2018), Bangalore, India, 19–18 August 2018. [Google Scholar]

9. Inmon, W.H.; Nesavich, A. *Tapping into Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence*; Pearson Education: London, UK, 2007. [Google Scholar]

10. Zafary, F. Implemetation of business intelligence considering the role of information systems integration and enterprise resource planning. *J. Intell. Stud. Bus.* **2020**, *1*, 59–74.

11. Pomsar, L.; Brecko, A.; Zolotova, I. Brief overview of edge ai accelerators for energy-constrained edge. In Proceedings of the 2022 IEEE 20th Jubilee World Symposium on

10.48047/jocaaa.2023.31.02.43

Applied Machine Intelligence and Informatics (SAMI), Poprad, Slovakia, 19–22 January 2022. [Google Scholar] [CrossRef]

12. Pääkkönen, P.; Pakkala, D. Extending reference architecture of Big Data Systems towards machine learning in edge computing environments. J. Big Data 2020, 7, 25. [Google Scholar] [CrossRef]

13. Bao, G.; Guo, P. Federated learning in cloud-edge collaborative architecture: Key Technologies, applications and challenges. J. Cloud Comput. 2022, 11, 94. [Google Scholar] [CrossRef]

14. Rong, G.; Xu, Y.; Tong, X.; Fan, H. An edge-cloud collaborative computing platform for building AIoT applications efficiently. J. Cloud Comput. 2021, 10, 36. [Google Scholar] [CrossRef]

15. Lalanda, P.; Hamon, C. A service-oriented edge platform for cyber-physical systems. CCF Trans. Pervasive Comput. Interact. 2020, 2, 206–217. [Google Scholar] [CrossRef]

16. Xu, R.; Jin, W.; Kim, D.H. Knowledge-based Edge Computing framework based on CoAP and HTTP for enabling heterogeneous connectivity. Pers. Ubiquitous Comput. 2020, 26, 329–344. [Google Scholar] [CrossRef]

17. Trakadas, P.; Masip-Bruin, X.; Facca, F.M.; Spantideas, S.T.; Giannopoulos, A.E.; Kapsalis, N.C.; Martins, R.; Bosani, E.; Ramon, J.; Prats, R.G.; et al. A reference architecture for cloud—Edge meta-operating systems enabling cross-domain, data-intensive, ML-assisted applications: Architectural overview and key concepts. Sensors 2022, 22, 9003. [Google Scholar] [CrossRef]

18. Lootus, M.; Thakore, K.; Leroux, S.; Trooskens, G.; Sharma, A.; Ly, H. A VM/containerized approach for scaling tinyml applications. *arXiv* **2022**, arXiv:2202.05057