FirstEigen White Paper

A Framework for AWS S3/Azure ADL/GCP Data Lake Validation: Overcome the Limitations of Deequ, Great Expectations and Other Rules-Based Approaches

FirstEigen™ | Gartner Cool Vendor | IDC Innovator

www.FirstEigen.com • contact@firsteigen.com

Executive Summary

Without effective and comprehensive validation, a Data Lake becomes a data swamp. With the accelerating adoption of AWS S3/Azure/GCP as the data lake of choice, the need for autonomously validating data has become critical. While solutions like Deequ, Griffin, Great Expectations provide the ability to validate AWS/Azure/GCP data, these solutions rely on rule-based approach that are rigid, non-flexible, static and not scalable for 100's of data assets and often prone to rules coverage issues. These also solutions do not provide an easy way to access audit trail of results.

Solution: A scalable solution that can deliver trusted data for tens of 1,000's of datasets has no option but to leverage AI/ML to autonomously track data and flag data errors. It also makes it an organic, self-learning system that evolves with the data.

Current Approach & Challenges

The current focus in AWS Data Lake projects is on data ingestion, the process of moving data from multiple data sources (often of different formats) into a single destination. After data ingestion, data is analyzed which is where data errors/issues begin to surface. Our research estimates that an average of 30-40% of any analytics projects is spent identifying and fixing data issues. In extreme cases, the project can get abandoned entirely.

Current data validation tools such as Deequ are designed to establish data quality rules for one bucket at a time—as a result there are significant cost issues in implementing these solutions for 100's of buckets. Bucket-wise focus often leads to incomplete set of rules or often not implementing any rules at all.

In general, data engineering team experiences the following operational challenges while integrating data validation solutions:

- Every dataset's underlying behavior is deeply analyzed and understood. Then the user derived properties are verified to be applicable for new versions of the dataset. This is critical as any application consuming the data must be able to rely on it. The reliability of future data is highly dependent on the user's ability to foresee and predict everything that could possibly go wrong.
- It is time consuming to analyze the data and consult the subject matter experts to determine what rules needs to be implemented.
- Implementation of the rules specific to each bucket. So, the effort is linearly proportional to the number of buckets in the S3 Data Lake
- Existing open-source tools comes with very limited audit trail capability. It's extremely difficult for the business to go back a week or a month and review past results or compare them to the present. Generating audit trail of the rule execution results for compliance requirements takes significant time and effort from the data engineering team.
- Maintaining the 100's of implemented rules (seen from the example in Exhibit-1), for 1000's of tables quickly adds up to unmanageable proportions. Its labor intensive and needs an army of Data Engineers and Subject Matter Experts (SME's) to support its future existence.

Deequ Example: To test for the following properties of a data set-
- There are at least 3 million rows in total.
- `review_id` is never NULL.
- `review_id` is unique.
- `star_rating` has a minimum of 1.0 and a maximum of 5.0.
- `marketplace` only contains "US", "UK", "DE", "JP", or "FR".
- `year` does not contain negative values.

This is the code needed using Deequ to reflect the previous statements.

```
import com.amazon.deequ.{VerificationSuite, VerificationResult}
import com.amazon.deequ.VerificationResult.checkResultsAsDataFrame
import com.amazon.deequ.checks.{Check, CheckLevel}

val verificationResult: VerificationResult = { VerificationSuite()
  // data to run the verification on
  .onData(dataset)
  // define a data quality check
  .addCheck(
    Check(CheckLevel.Error, "Review Check")
      .hasSize(_ >= 3000000) // at least 3 million rows
      .hasMin("star_rating", _ == 1.0) // min is 1.0
      .hasMax("star_rating", _ == 5.0) // max is 5.0
      .isComplete("review_id") // should never be NULL
      .isUnique("review_id") // should not contain duplicates
      .isComplete("marketplace") // should never be NULL
      // contains only the listed values
      .isContainedIn("marketplace", Array("US", "UK", "DE", "JP", "FR"))
      .isNonNegative("year")) // should not contain negative values
  // compute metrics and verify check conditions
  .run()
}

// convert check results to a Spark data frame
val resultDataFrame = checkResultsAsDataFrame(spark, verificationResult)
```

Exhibit 1: Sample of code needed to operationalize very elementary checks in Deequ for a single dataset. It's extensive and laborious to execute for 1000's of datasets

Source: https://aws.amazon.com/blogs/big-data/test-data-quality-at-scale-with-deequ/)

Solution Framework

Organizations must consider validation solutions that, at minimum, meet the following criteria:

1. **Machine Learning Enabled:** A rule-based system implementation will not scale. Only machine learning systems can scale to the level required by enterprises dealing with large data volumes. Solutions must leverage AI/ML to:
   - Identify and codify the data fingerprint for detecting data errors related to Freshness, **Completeness, Consistency, Conformity, Uniqueness, and Drift**.
   - Effort required for establishing validation checks should not depend on the number of S3 Buckets. Ideally, Data Engineer should be able to establish validation checks for 100s buckets with a single click.
2. **Autonomous:** Solution must be highly intelligent and must be able to:
   - Establish validation checks autonomously when a new data bucket is created (Exhibit-2).
   - Update and organically evolve the existing validation checks autonomously when the underlying data evolves.
   - Perform validation on the incremental data as soon as the data arrives.
3. **Data Variety Support**: Solutions must be able to parse a variety of data formats without the need for transformations.
   - Parquet, Orc, JSON, EDI, Flat files (CSV, PSV, TSV), etc.
4. **Scalability:** Solutions must offer the same level of scalability as the underlying Big Data platform used for storage and computation. Any scalable system must perform most of its operations automatically. Hence scalability inherently needs the subtle guiding hand of AI/ML to be able to validate tens of 1,000's of data sets.

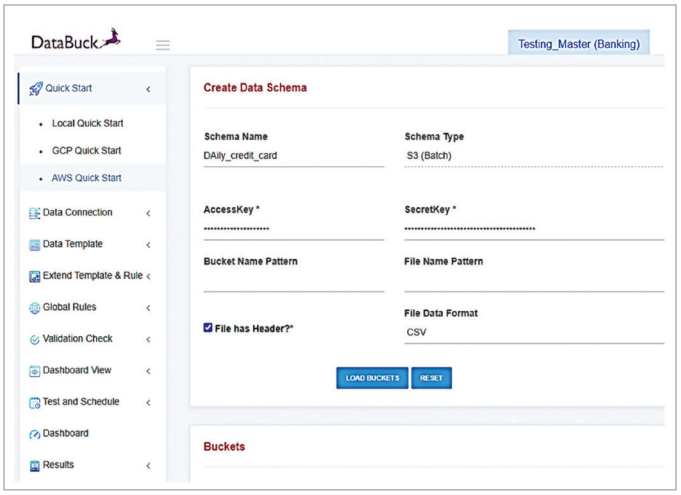

Exhibit 2: Scalable data validation solution must be autonomous, operationalized without any coding and by just pointing at the data bucket

5. **In-Situ:** Solutions must validate data at source and not pull the data to a centralized location. This avoids latency and security risks. This also minimizes the hardware needed and reduces costs.
6. **Serverless:** Solutions must provide a serverless scalable data validation engine. Ideally, solution must be using the underlying infrastructure available for the data ingestion and transformation such as AWS Glue.
7. **Part of the Data Validation Pipeline:** Solution must be easily integrated as part of the data pipeline jobs. Data flow must be controlled automatically in response to good or bad data and manage its spread throughout the rest of the organization.
8. **Integration and Open API:** Solutions must provide open APIs for easy integration with the enterprise scheduling, workflow, and security systems.
9. **Audit Trail/Visibility of Results:** Solutions must provide easy to navigate audit trail of the validation test results (Exhibit-3).
10. **Business Stakeholder Control:** Solutions must provide business stakeholders full control of the implemented rules. Business stakeholders should be able add/modify/deactivate rules without involving data engineers.

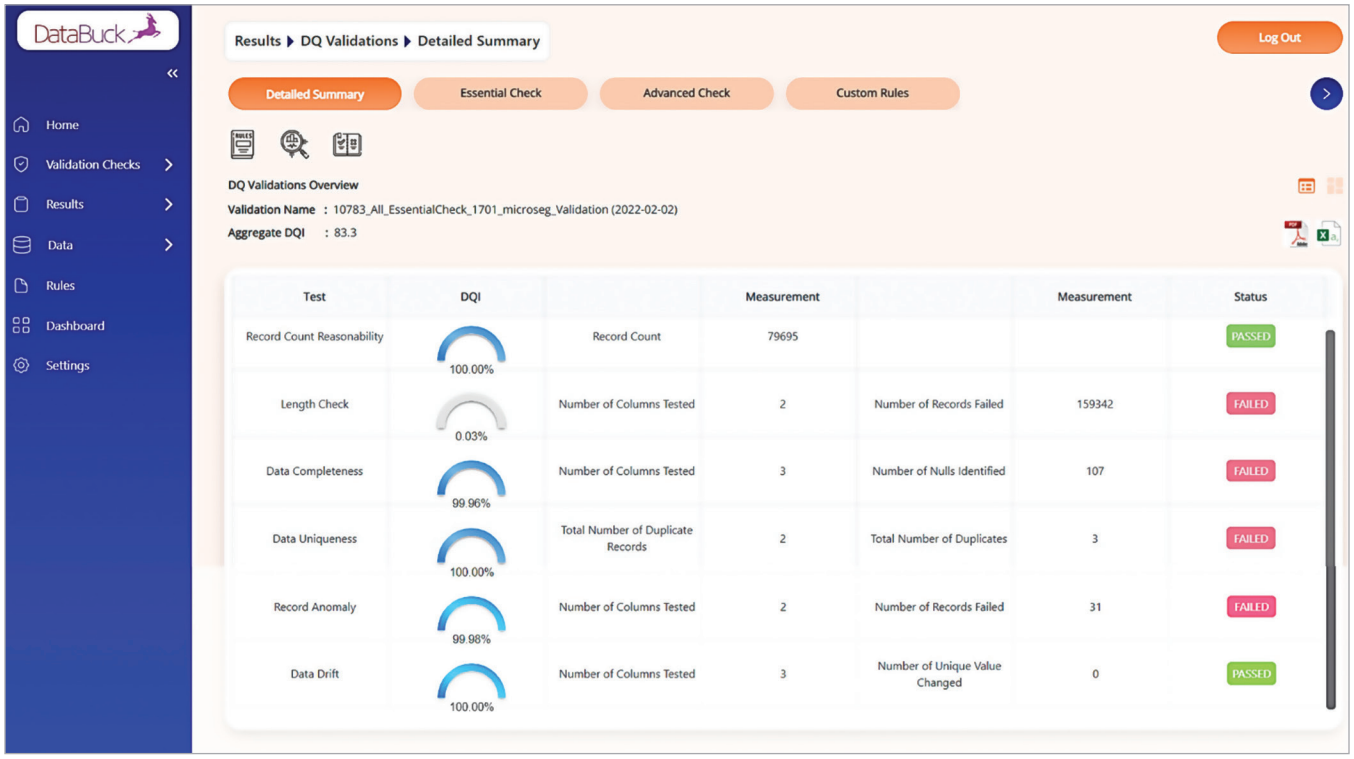

Exhibit 3: Example of a business stakeholder friendly dashboard that is easy to audit historical data quality.

Conclusion

Data is the most valuable asset for modern organizations. Current approaches for validating data, in particular AWS S3 Data Lake, are full of operational challenges leading to trust deficiency, time-consuming, and costly methods for fixing data errors. There is an urgent need to adopt a standardized autonomous approach for validating the AWS S3 data to ensure prevent data lake from becoming data swamp.

DataBuck - Serverless, Autonomous, In-Situ Data Validation

FirstEigen's DataBuck is recognized by Gartner and IDC as the most innovative data validation software for the Lake and the Cloud. By leveraging AI/ML it's >10x more effective in catching unexpected data errors in comparison to traditional DQ tools. DataBuck's out-of-the-box 9-Point Data Quality checks needs no coding, minimal human intervention, and is operationalized in just a few clicks. It increases the scalability of discovering and applying essential data quality checks to 1,000's of tables by auto-discovering relationships and patterns, auto updating the rules, and continuously monitoring new incoming data. It reduces man-years of labor to days.

DataBuck

DataBuck – Benefits

People Productivity Boost >80%

Reduction in Unexpected Errors: 70%

Cost Reduction >50%

Time Reduction to Onboard Data Set ~90%

Increase in Processing Speed >10x

Cloud Native

FirstEigen™ | Gartner Cool Vendor | IDC Innovator