[C]OMPLERE

Business Rule Validator
Automate, Monitor, and Trust Your Data

Purpose

Unlocktrusted, high-quality data with the Business Rule Validator Framework. Automate daily SQL-based validations, track incremental changes, and monitor rule execution with powerful dashboards.

Why Data Quality Matters?

Business Impact
Poordatacausesofbusinessdecisions to fail.

Consequences
Resultsinrevenuelossandrisks regulatory non-compliance

Customer Trust
Leadstocustomerdissatisfaction and erodes trust.

7 Key Dimensions of Data Quality

Accuracy
Correct and precise values.

Completeness
No missing information.

Uniqueness
Eliminate duplicates.

Timeliness
Up-to-date data.

Validity
Correct format and range.

Consistency
Uniform across sources.

Integrity
Reliable relationships.

Why Data Quality Framework?

Structured Approach
A repeatable method for data excellence.

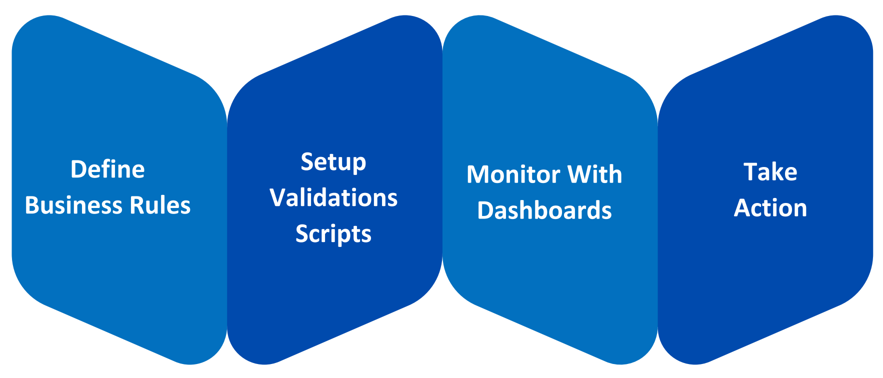
Key Attributes
Ensuresdataiscomplete,accurate, accurate, and consistent.

Actionable Data
Providestimelyandactionable insights.

Severity of Data Quality

Critical	**High**
Medium	**Low**

DQF - Step by Step Process



Why Use It?

✓ Improve data quality
✓ Build trust in business data
✓ Catch issues early
✓ Enable data-driven decision-making

Key Features

📅 **Automated Daily Validation**
🔍 **SQL-Based Business Rule Checks**
🕒 **Incremental Data Validation**
🧩 **Parameterised SQL Scripts**
📊 **Dashboard for Monitoring Trends**

Benefits to Business

🔔 Continuous Monitoring
⚡ Faster Issue Resolution
📈 Consistent Data Standards
🤝Stronger Data Governance

Why It Matters?

Build trust in **your data**
Enable faster **root cause analysis**
Ensure **seamless operations** across systems

Validate more intelligently. Keep a closer watch. Foster trust in your data starting today.

Data Parity Checks vs Business Rule Checks

Know the Difference

Data Parity Checks

Purpose

- Ensure data consistency between source and target systems (e.g., after ETL or replication)
- Confirm records and attributes match exactly

Example

- Comparing OrderAmount in source vs OrderAmount in the warehouse
- Checking row counts between systems

Key Question

Is the data identical in both systems?

Business Rule Checks

Purpose

- Validate data against predefined business logic or expectations
- Ensure data meets quality thresholds for use in analysis or operations.

Example

- OrderAmount > 0
- CustomerDOB should not be in the future
- Mandatory fields are populated

Key Question

Does the data comply with business policies and quality standards?

Check Type	Goal	Example Validations
Data Parity Checks	Consistency between systems	Source vs Target match
Business Rule Checks	Compliance with business expectations	Field must be positive

Why Both Matters?

- Parity checks ensure trust in data pipelines
- Business rules ensure trust in the data itself

Why It Matters?

- Build trust in **your data**
- Enable faster **root cause analysis**
- Ensure **seamless operations** across systems

Validate your pipelines = Validate your business.

Use both Data Parity & Business Rule checks for complete data trust.

[C]OMPLERE

Business Rule Validator
Know how to Setup?

Define Rules

1

- Start with Business Objectives
- Identify and define rules for various departments, including customer, product, financial, and operational data.
- Build rules around 7 Dimensions
- Define Rule Logic Clearly
- Define Rule Severity
- Define Thresholds

```
{
 "TEAM_NAME": "Bakehouse",
 "DOMAIN_NAME": "Transactions",
 "RULE_CATEGORY_NAME": "BUSINESS_RULE_CHECK",
  "RULES": [{
         "RULE_ID": 1,
         "RULE_NAME": "unique_transactions_id_check",
         "RULE_CATEGORY":"Uniqueness",
         "SEVERITY":"Critical",
         "FAIL_SQL": "/sql/rules_sql/unique_transactionId_fail.sql",
         "PASS_SQL":"/sql/rules_sql/unique_transactionsId_pass.sql",
         "TABLES_CHECKED": "sales_transactions",
         "INVENTORY": "Transactions Validator Rules",
         "COMMENTS": "Check if transactionID is unique.",
         "PASS_THRESHOLD": 100,
         "BOOKMARK_START_DATE":"2025-04-10",
         "DEFAULT_BOOKMARK_START_DATE":"2025-03-20"

        }
 ]
}
```

```
{
 "TEAM_NAME": "DATA",
 "RULE_CATEGORY_NAME": "DATA_PARITY_CHECK",
 "DOMAIN_NAME":"DATA",
 "RULES": [
      {
        "RULE_ID": 1,
        "RULE_NAME": "DEMO-EMP-DATA-CHECK",
        "SOURCE_SQL": "/sql/dpc_sql/source_transactions.sql",
        "TARGET_SQL": "/sql/dpc_sql/target_transactions.sql",
        "JOIN_DIMENSIONS": "transactionID",
        "INVENTORY": "Demo Tables",
        "TABLE_NAME": "Sales_transactions",
        "COMMENTS": "Running only Demo tables",
        "METRIC_DIMENSIONS": "transactionID",
        "THRESHOLD": 95,
        "RECORD_THRESHOLD":95
      }
 ]
}
```

Defining Validations

2

- Null Checks
- Format Checks
- Range Checks
- Uniqueness Checks
- Referential Integrity Checks
- Value Checks / Domain Constraints
- Timeliness Checks
- Source & Target Comparision

```
## TransactionID Unique Fail Test Script

SELECT transactionID, COUNT(*) FROM transactions GROUP BY
transactionID HAVING COUNT(*) > 1;
```

```
## TransactionID Unique Pass Test Script

SELECT transactionID, COUNT(*) FROM transactions GROUP BY
transactionID HAVING COUNT(*) = 1;
```

```
## Source SQL
SELECT transactionID, customerID, quantity, unitPrice, totalPrice,
paymentMethod, cardNumber
FROM samples.bakehouse.sales_transactions
```

```
## Target SQL
SELECT transactionID, customerID, quantity, unitPrice, totalPrice,
paymentMethod, cardNumber
FROM test_data.bakehouse.sales_transactions
```

Executing Rules

3

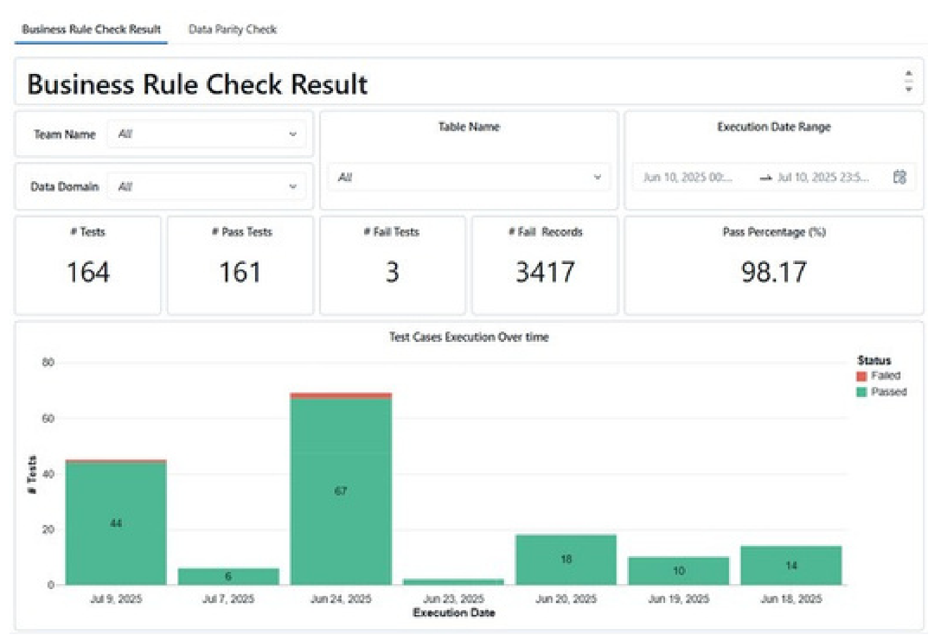
- Easily define rules and validations—no complex setup required The
- solution auto-generates the full architecture behind the scenes
- Rules are executed automatically across your datasets Scalable,
- repeatable, and aligned with business needs Designed for both tech
- and non-tech teams to use with ease

```
config = load_config(file_path)
result = check_business_rules(config,spark,dbutils)
print("Business Rule Validator Result:")
print(json.dumps(result, indent=4))
```

Monitoring Data Quality:

4

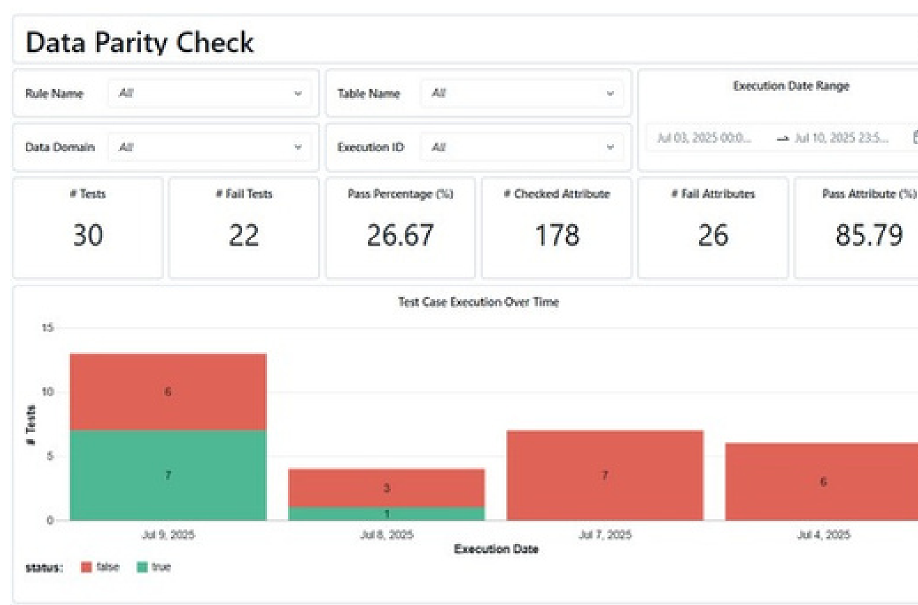
- Tracks the number of tests passed vs tests failed each day
- Helps monitor the daily health of your data pipelines
- Identifies critical failures that need immediate attention
- Enables proactive issue resolution before data Impacts business
- Builds accountability through visible quality metrics
- Supports continuous improvement of validation rulesover time



Business Rule Check Result

Team Name: All
Data Domain: All
Table Name: All
Execution Date Range: Jun 10, 2025 00:... → Jul 10, 2025 23:5...

# Tests	# Pass Tests	# Fail Tests	# Fail Records	Pass Percentage (%)
164	161	3	3417	98.17



Data Parity Check

Rule Name: All
Table Name: All
Data Domain: All
Execution ID: All
Execution Date Range: Jul 03, 2025 00:0... → Jul 10, 2025 23:5...

# Tests	# Fail Tests	Pass Percentage (%)	# Checked Attribute	# Fail Attributes	Pass Attribute (%)
30	22	26.67	178	26	85.79

Get full visibility into your data pipelines, catch mismatches early, and build trust in your data—every day!