Assessing Data Quality in Relational Database Migration: The Data Quality Score (DQS)

Abhinav Srivastava
Gemini Solutions
St. Louis, MO, USA
abhi9.usa@gmail.com

Tanya Chaudhary
Gemini Solutions
Gurugram, India
tanya.chaudhary010704@gmail.com

Abstract

Relational database migration is a high-risk operation in which data quality defects frequently evade conventional validation techniques such as row-count reconciliation, null-rate checks, and schema comparison. This paper presents the *Data Quality Score* (DQS), a reference-based, mathematically transparent framework for post-migration validation of relational datasets. DQS evaluates a target dataset against a source reference across six formally defined dimensions: Row Fidelity (RF), Accuracy (A), Integrity (I), Structural Fidelity (SF), Consistency (K), and Business Rule Compliance (B). A key contribution is RF, a harmonic-mean metric that jointly captures missing and spurious records without double-penalisation. Dimensions are aggregated via a risk-aware weighted average into a single 0–100 score with a formal weight-redistribution mechanism for inapplicable dimensions. The framework is explicitly scoped to comparison-based scenarios—database migration and ETL regression testing—where a reference dataset is naturally available. Empirical evaluation on four production healthcare datasets (millions of rows, more than 800 pipeline runs) shows a strong negative correlation ($r < -0.9$, $p < 0.001$) between DQS and downstream data-readiness defects, demonstrating both a reliable go/no-go index for migration acceptance and a diagnostic breakdown for identifying defect sources.

VLDB Workshop Reference Format:
Abhinav Srivastava and Tanya Chaudhary. Assessing Data Quality in Relational Database Migration: The Data Quality Score (DQS). VLDB 2026 Workshop: 15th International Workshop on Quality in Databases (QDB'26).

1 Introduction

Relational database migration—the movement of tables, rows, and constraints between systems—is pervasive in enterprise data engineering: system upgrades, on-premises-to-cloud transitions, platform consolidations, and regulatory re-platforming all produce migration pipelines. Despite mature tooling, migration projects carry substantial data quality risk: records go missing, primary-key semantics change, transformation rules introduce cell-level errors, and referential constraints are silently violated [1, 2].

Conventional validation is fragmented. Row-count reconciliation detects only complete table loss. Null-rate checks miss value-level corruption. Schema diffing cannot detect silent data loss within structurally preserved tables. None produces a unified, automated go/no-go decision for migration release.

DQS scope. DQS is a reference-based quality score designed explicitly for relational database migration and ETL regression validation, where a source dataset is naturally available and defines ground truth. In migration scenarios, the source system snapshot is the canonical ground truth—DQS does not assume an external "perfect" dataset. This constraint is a feature: it bounds each metric to scenarios where semantics are well-defined.

Why a single composite score? A scalar index is operationally necessary: automated go/no-go gating requires a threshold; data governance stakeholders must interpret quality reports; regression tracking across runs requires a comparable index. DQS provides this index while preserving diagnostic transparency through its component vector (RF, A, I, SF, K, B). As AI pipelines increasingly rely on relational data migration for RAG corpora and LLM fine-tuning datasets, DQS provides a principled validation gate before AI ingestion.

Contributions:

- A unified DQS framework aggregating six dimensions into a single 0–100 score scoped to migration validation.
- *Row Fidelity* (RF): a novel harmonic-mean metric jointly capturing structural completeness and noise without double-penalisation.
- A formal weight-redistribution mechanism preserving score validity when dimensions are inapplicable.
- Empirical validation ($r < -0.9$, $p < 0.001$; more than 800 pipeline runs) on four production healthcare datasets.
- A two-level gating mechanism: composite threshold blocking (DQS < 70) with component-level review flags (any dimension < 75), demonstrated across more than 800 pipeline runs at an enterprise deployment.

2 Related Work

Rahm and Do [3] survey data cleaning problems arising during data integration and migration, establishing the taxonomy of defect types this paper addresses. Abedjan et al. [4] provide a comprehensive survey of data profiling; DQS extends profiling into comparison-based post-migration validation. Pipino et al. [5] catalogued quality metrics as simple ratios and weighted aggregations but address neither structural mismatch nor inapplicable dimensions. Chug et al. [6] propose a PCA-based composite score producing opaque latent values requiring domain-specific re-fitting. DQSOps [7] focuses on operational data quality scoring and pipeline monitoring rather than on source–target structural comparison of the kind required for migration validation. ISO/IEC 25012 [8] defines

Proceedings of the VLDB Endowment. ISSN 2150-8097.

a quality vocabulary without formulas or composite scoring. Recent overviews of data quality assessment [9] confirm that systematic, dimension-level assessment remains an open problem. To the best of our knowledge, no existing framework simultaneously combines (i) harmonic-mean structural fidelity, (ii) duplicate-pattern preservation, (iii) formal weight redistribution, and (iv) a composite score explicitly scoped to migration validation. Composite single-score indices have proven effective in adjacent domains—for instance, the Application Stability Index (ASI) aggregates multiple signals into one interpretable score for assessing software application stability [10]—but no analogous index targets relational migration validation.

Table 1 highlights that DQS combines transparent composite scoring, structural row-level comparison, duplicate PK pattern preservation, formal weight redistribution, and explicit migration scope validation in a single unified instrument.

Table 1: Feature comparison of DQS against existing frameworks. F1=Composite score, F2=Row structure check, F3=Duplicate PK preservation, F4=Weight redistribution, F5=Migration scope.

Framework	F1	F2	F3	F4	F5
ISO/IEC 25012 [8]	No	No	No	No	No
Pipino et al. [5]	Yes	No	No	No	No
Chug et al. [6]	PCA	No	No	No	No
DQSOps [7]	No	No	No	No	Part.
Taleb et al. [11]	Part.	No	No	No	No
DQS (ours)	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

2.1 Relation to Classical Data Quality Dimensions

To position DQS within the broader data quality literature, Table 2 maps each DQS dimension to its closest counterpart in the classical taxonomies of Wang and Strong [12] and ISO/IEC 25012 [8]. Most DQS dimensions specialise a classical concept to the migration setting, where quality is defined *relative to a source* rather than in absolute terms. Structural Fidelity has no classical counterpart: preservation of duplicate-count patterns is meaningful only in comparison-based scenarios and is, to our knowledge, novel to this work. This mapping frames DQS as a migration-pivoted instantiation of established data quality theory rather than an unrelated measurement scheme.

3 The DQS Framework

DQS evaluates a target dataset T against source S across six dimension scores in $[0, 100]$, aggregated into DQS $\in [0, 100]$. Figure 1 shows the computation pipeline. Table 3 defines key symbols.

Schema correspondence. DQS assumes that schema correspondence between S and T is established before scoring: either the schemas are identical, or an explicit column mapping is supplied by the migration specification (the standard artefact of any migration project). All cell-level quantities, including C_{npk}, are computed over the mapped column pairs. Automated schema matching itself is a distinct, well-studied problem [3] and is out of scope for DQS.

Table 2: Mapping of DQS dimensions to classical data quality taxonomies.

DQS	Wang & Strong [12]	ISO/IEC 25012 [8]
Row Fidelity (RF)	Completeness	Completeness
Accuracy (A)	Free-of-error	Accuracy
Integrity (I)	Conformance	Compliance
Struct. Fid. (SF)	*(novel)*	*(novel)*
Consistency (K)	Repr. consistency	Consistency
Bus. Rule (B)	Contextual fitness	Credibility

Table 3: Key symbols used in DQS formulas.

Symbol	Definition
$N_s,\ N_t$	Row *sets* of source S and target T; cardinality $\lvert N_s\rvert, \lvert N_t\rvert$
N_m	Set of rows whose PK appears in both S and T
C_{npk}	Non-PK columns comparable in S and T
C_{tot}	Total non-PK cells: $\lvert N_m\rvert \times C_{npk}$
C_{mis}	Non-PK cells where $S \neq T$ (matched rows)
$d_s(k), d_t(k)$	PK value k occurrence count in S, T
E_{dup}	$\sum_k \lvert d_s(k) - d_t(k)\rvert$ (PK deviation)
C_{fmt}	Cells violating transformation constraints
C_{vr}, C_{tr}	Cells satisfying / evaluated for business rules
$w_j,\ w'_j$	Original and redistributed weight, dimension j

3.1 Row Fidelity (RF)

RF measures structural row preservation using PK identity. Completeness and noise ratios:

$$C_r = |N_m| \,/\, |N_s| \tag{1}$$

$$N_r = |N_m| \,/\, |N_t| \tag{2}$$

Intuitively, C_r is analogous to *recall*: the fraction of source rows recovered in the target. N_r is analogous to *precision*: the fraction of target rows that legitimately originate from the source. RF is their harmonic mean, expressed as a percentage—the *F1-score of row preservation*:

$$\mathrm{RF} = \frac{2 \cdot C_r \cdot N_r}{C_r + N_r} \times 100 \tag{3}$$

The harmonic mean penalises large asymmetries: a complete but noisy dataset (C_r=1, $N_r \approx 0$) receives RF≈0. Edge cases: both sets empty ⇒ RF = 100; one empty ⇒ RF = 0.

3.2 Accuracy (A)

A measures non-PK attribute value fidelity for matched rows, separating content defects from structural ones captured by RF:

$$A = \left(1 - \frac{C_{mis}}{C_{tot}}\right) \times 100 \tag{4}$$

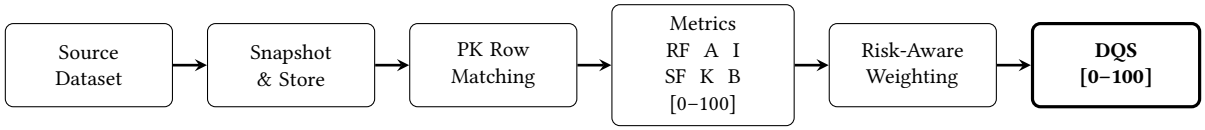

Figure 1: DQS computation pipeline: Source → Snapshot → PK Matching → Metrics → Weighting → DQS.

For sparse tables where non-PK columns have varying presence, C_{tot} counts all schema-defined non-PK cells for matched rows N_m, treating absent values as NULL—consistent with how NULL is handled by both the Accuracy and Integrity dimensions. If $C_{npk} = 0$ (PK-only tables), A is undefined and its weight is redistributed per Section 3.6.

3.3 Integrity (I)

I validates m structural constraints (NOT NULL, data type, allowable value) in the target DDL. Let $C_{valid}^{(j)}$ be the number of cells satisfying constraint j, and $C_{tot}^{(j)}$ the total cells evaluated for constraint j—that is, the count of non-null cells in the column governed by constraint j across all matched rows N_m:

$$I = \frac{1}{m} \sum_{j=1}^{m} \frac{C_{valid}^{(j)}}{C_{tot}^{(j)}} \times 100 \tag{5}$$

3.4 Structural Fidelity (SF)

SF measures preservation of duplicate-count patterns for each PK value. SF is a *fidelity* metric, not a uniqueness metric in the sense of ISO/IEC 25012 [8] or Wang and Strong [12]: a perfect SF score indicates the target faithfully replicates the source's duplicate structure, not that records are unique. In the degenerate case where $E_{dup} > |N_s|$, SF is clamped to 0, indicating complete structural breakdown.

$$E_{dup} = \sum_{k \in PK} |d_s(k) - d_t(k)| \tag{6}$$

$$\text{SF} = \max\left(0,\ \left(1 - \frac{E_{dup}}{|N_s|}\right) \times 100\right) \tag{7}$$

3.5 Consistency (K) and Business Rule Compliance (B)

K measures adherence to transformation constraints (rounding, date formats, unit conversions):

$$K = \left(1 - \frac{C_{fmt}}{C_{tot}}\right) \times 100 \tag{8}$$

B measures satisfaction of domain-specific multi-column predicates, where C_{tr} is the number of rows evaluated against defined business rule predicates and C_{vr} the number satisfying all predicates:

$$B = \frac{C_{vr}}{C_{tr}} \times 100 \tag{9}$$

3.6 Risk-Aware Weighting and Redistribution

Each dimension has a Defect Severity Index S_j (1=Minor, 2=Major, 3=Critical) and a Failure Propagation Index P_j (1=Low, 2=Medium, 3=High). Normalised weight:

$$w_j = \frac{S_j + P_j}{\sum_{k=1}^{6} (S_k + P_k)} \tag{10}$$

Table 4 presents the full derivation. When a dimension is inapplicable, define the total skipped weight and redistribute proportionally among remaining dimensions Ω:

$$w_{\text{skip}} = \sum_{j \notin \Omega} w_j \tag{11}$$

$$w_i' = \frac{w_i}{1 - w_{\text{skip}}}, \quad \forall\, i \in \Omega \tag{12}$$

Equation (12) guarantees $\sum_{i \in \Omega} w_i' = 1$ whenever $w_{\text{skip}} < 1$. Final score:

$$\text{DQS} = \sum_{i \in \Omega} w_i' \cdot Q_i \tag{13}$$

Table 4: Risk-aware weight derivation (sum = 1.000).

Dimension	S_j	P_j	S_j+P_j	w_j
Row Fidelity (RF)	3	3	6	0.240
Accuracy (A)	3	3	6	0.240
Integrity (I)	2	2	4	0.160
Structural Fidelity (SF)	2	2	4	0.160
Consistency (K)	1	2	3	0.120
Bus. Rule Compliance (B)	1	1	2	0.080
Total			**25**	**1.000**

The six dimensions are not claimed to be exhaustive: they cover the defect classes observed in relational migration practice. The weighting scheme is deliberately extensible—a seventh dimension (e.g., timeliness/freshness for change-data-capture scenarios) can be added by assigning its S_j and P_j indices and renormalising Equation (10); no other formula changes are required.

3.7 Weight Justification

The S_j and P_j values are informed by established treatments of data quality dimensions, notably Batini and Scannapieco [1], and by the well-documented risk of data defects propagating downstream through dependent systems [2]; the specific index values are assigned by the authors based on observed migration-defect behaviour. RF and A receive S_j=3 because missing or corrupted rows propagate irreversibly into downstream systems and are the hardest class of migration defect to detect post-deployment. I and SF receive S_j=2 as violations, while serious, are detectable through schema validation and row-count audits after migration. K and B receive $S_j \leq 2$ as they affect derived transformation logic rather than core record fidelity, and their impact is typically bounded to specific calculations rather than propagating system-wide.

3.8 Score Interpretation and Semantics

DQS is not a universal quality scalar. Within its defined scope it measures migration fidelity along six orthogonal dimensions. Two datasets with equal DQS may differ in component composition; DQS should be interpreted as a decision-support index complemented by its component vector. Table 5 maps composite score ranges to migration acceptance decisions. Any individual component score below 75 triggers a mandatory human review flag regardless of the composite DQS value, providing a two-level gating mechanism.

The thresholds in Table 5 were calibrated empirically against several months of defect logs from the deployment described in Section 5: the automatic blocking threshold (70) was chosen as the highest value at which every run that subsequently produced a severity-1 or severity-2 data-readiness ticket fell below it, and the per-component review threshold (75) was chosen to capture single-dimension failures of the kind exhibited by dataset D1 (a Good composite tier with one dimension falling in the Acceptable tier). Both thresholds are configuration parameters and can be recalibrated per organisation from local defect history.

Table 5: DQS migration acceptance guidance. Automated pipeline blocking applies when DQS < 70 or any component < 75.

DQS	Status	Decision
90–100	Excellent	Approved for downstream use
75–89	Good	Minor defects; document & monitor
70–74	Acceptable	Component review; not auto-blocked
60–69	Poor	Auto-blocked; remediate before release
<60	Critical	Blocked; escalate to data governance

3.9 Co-occurring Defects

Real migrations frequently exhibit multiple defect types simultaneously. Because the six dimensions are computed independently over largely disjoint defect classes, co-occurring defects degrade multiple components at once, and the composite DQS drops faster than any single conventional check would indicate. Dataset D1 (Section 5) is itself a co-occurrence case: row loss and duplicate-pattern change degrade Row Fidelity and Structural Fidelity simultaneously while leaving null rates and schema checks untouched. The additive aggregation of Equation (13) treats dimension degradations as independent contributions; modelling interaction effects between defect types (e.g., row loss that also masks accuracy errors in the lost rows) is left to future work.

3.10 Design Rationale

3.10.1 Why a Reference Dataset? A natural objection is: why require a reference dataset at all? In database migration, the definition of correctness *is* the source system. There is no domain-independent ground truth: a null rate of 12% is acceptable if the source had 12%, and a defect if the source had 0%. Reference-free profiling can detect absolute anomalies but cannot detect the defining failure mode of migration: silent data loss, silent value corruption, and structural reorganisation. These defects are invisible to any method observing only the target.

The reference constraint is therefore a **prerequisite of correctness** in migration contexts, not a limitation. The source system snapshot is a natural artefact of every migration project. DQS is designed for the common case in which a source snapshot is available in migration pipelines, rather than the edge case where it is not.

3.10.2 Why Not a Machine Learning Approach? Three properties argue against a purely ML-based approach. **First, labelled training data does not exist.** A supervised model requires historical (migration output, defect label) pairs. Organisations rarely have enough labelled runs, particularly for rare high-severity defects such as silent row loss. DQS is fully operational from the first migration run.

Second, auditability is a hard requirement. In regulated industries such as healthcare, finance, and pharmaceuticals, migration acceptance must be fully traceable. A neural network score of 0.73 provides no such trail. DQS scores decompose to specific row counts, cell-level mismatches, or constraint violations, satisfying non-negotiable audit requirements.

Third, ML models conflate source–target comparison with statistical anomaly detection. An autoencoder trained on historical target tables flags distributions differing from training data—not from *this specific source dataset*. DQS performs direct pair-wise comparison, the correct semantics for migration validation. DQS and ML are therefore complementary: DQS for immediate post-migration acceptance gating; anomaly detection for ongoing operational monitoring.

4 Baseline Comparison

Conventional validation relies on three independent checks (declarative frameworks such as Great Expectations [13] extend these with column-level expectations but still evaluate datasets in isolation without source–target structural comparison): (i) *row-count reconciliation*—detects complete table loss only; (ii) *null-rate validation*—misses value corruption when non-null values change; (iii) *schema comparison*—no guarantee of data preservation within an unchanged structure.

Table 6: Detection capability for a representative structural defect (silent row loss with schema change). c1=Row Loss, c2=Duplicate Change, c3=Value Errors, c4=Root Cause. Partial = detects only when non-null values fall outside historical column range.

Method	c1	c2	c3	c4
Row-count	No	No	No	No
Null-rate	No	No	Partial	No
Schema diff	No	No	No	No
Gr. Expectations	No	No	Partial	No
DQS	**Yes (RF)**	**Yes (SF)**	**Yes (A,I,K)**	**Yes**

Table 6 compares detection capability against Dataset D1 (Section 5), which suffered migration-induced silent row loss during a schema normalisation step. Row count appeared consistent—duplicates partially compensated for lost rows—so row-count reconciliation reported no anomaly. Null-rate and schema checks also passed. DQS detected the defect through the Row Fidelity component (Acceptable tier) and Structural Fidelity capturing changes in duplicate PK patterns. Conventional approaches are dimension-specific and non-compositional, making them insufficient for defects spanning multiple dimensions simultaneously. DQS integrates all six dimensions into a single interpretable score while preserving component-level transparency for root-cause localisation, enabling both automated gating and targeted remediation. Unlike many rule-based data quality tools that evaluate datasets in isolation, DQS explicitly models source–target structural comparison and aggregates results into a unified decision-support score.

5 Enterprise Validation

5.1 Deployment Context

DQS has been operationally deployed over a multi-month period at a large healthcare enterprise migrating an on-premises relational database system to a cloud-hosted target platform. The full migration corpus spans over several thousand production tables across multiple business domains. The framework integrates with the enterprise ETL tool: at each run it snapshots source and target tables, computes all six DQS dimensions, and publishes composite scores to a data quality dashboard. The deployment uses two-level gating: datasets with DQS < 70 are automatically blocked from downstream pipelines; datasets with any individual component score below 75 trigger a mandatory human review flag regardless of the composite value.

5.2 Datasets

DQS was applied to all pipeline executions across the active migration batch. Results are reported in detail for four representative datasets selected to span the full complexity range—from simple reference lookups to complex transactional tables with composite PKs: **D1**—a large transactional table with composite PKs (millions of rows; schema normalisation defect). **D2**—an entity registry table (hundreds of thousands of rows; identifier format migration). **D3**—a time-series measurement table (millions of rows; temporal format normalisation). **D4**—a reference lookup table (hundreds of thousands of rows; low-complexity baseline).

5.3 Results

Table 7 presents the acceptance tier profile for each dataset. D1's composite DQS falls in the Good tier, so the pipeline was not automatically blocked. However, its Row Fidelity component falls in the Acceptable tier—below the per-component review threshold—triggering a mandatory manual review. This is the intended two-level gating behaviour: the composite score governs automated release decisions, while individual component scores below the review threshold escalate to human review regardless of the composite. A conventional null-rate-and-schema check would have raised no alert at either level. The Row Fidelity signal directed engineers to the row-matching layer, revealing that records were silently dropped during schema normalisation—a defect that would have propagated into downstream systems.

Table 7: DQS acceptance tier profiles across four representative datasets. Tiers: E=Excellent (≥90), G=Good (75–89), A=Acceptable (70–74), P=Poor (<70). *D1 composite tier is Good but RF falls in the Acceptable tier, triggering a component-level review flag.

Component	D1	D2	D3	D4
Row Fidelity (RF)	A*	G	G	E
Accuracy (A)	G	E	E	E
Integrity (I)	E	E	E	E
Structural Fidelity (SF)	G	G	E	E
Consistency (K)	G	G	G	E
Bus. Rule (B)	E	E	E	E
DQS tier	**G***	**G**	**E**	**E**
Decision	*Review*	*Pass*	*Pass*	*Pass*

Across more than 800 pipeline runs covering several thousand production tables over a multi-month period, the Pearson correlation—computed across these runs, with each run contributing a (DQS at pipeline completion, data-readiness defect count) pair (Jira tickets within 48 h)—was $r < -0.9$ ($p < 0.001$). Of all pipeline runs that subsequently generated data-readiness Jira tickets, no run had DQS ≥ 70 at completion—zero false negatives *within the 48-hour detection window* were observed across the entire evaluation period. Quality triage time decreased by an estimated ≈60%, based on team self-assessment comparing average investigation time per defect before and after DQS deployment, as stewards could direct investigation to the specific failing component. These results indicate that DQS not only detects defects but also prioritises investigation by isolating the responsible quality dimension.

5.4 Pipeline Integration

DQS is defined as a tool-agnostic algorithm: its computation depends only on access to the source and target datasets, not on any particular platform. The six metrics are computed independently and in parallel, making DQS scalable to large tables without blocking pipeline execution. Table 8 lists illustrative integration patterns showing how DQS can be embedded into common enterprise toolchains—any of these, or any equivalent tool, can host the computation. In all patterns, DQS scores are written to a central quality metadata store enabling longitudinal tracking across migration runs.

5.5 Threats to Validity

Limitations: results reflect one healthcare enterprise; cross-domain generalisation requires independent replication. The DQS formulation itself is domain-agnostic—no formula depends on healthcare-specific semantics—so replication in other domains requires only source and target datasets from those domains. The multi-month window covers one migration cycle; across different domains the correlation magnitude may vary, however the directional relationship between DQS and downstream defects is expected to remain strongly negative. The Jira-ticket detection window of 48 h may

Table 8: Illustrative integration patterns showing how the tool-agnostic DQS algorithm can be embedded into common enterprise toolchains. These are examples, not requirements; any equivalent tool can host the computation.

Toolchain	Integration Pattern
Enterprise ETL Tool	Post-load validation session; DQS invoked after each pipeline run; scores to dashboard DB
Azure Data Factory	DQS as Azure Function; triggered by pipeline completion via Event Grid
Apache Spark / Databricks	DQS as Spark DataFrame UDF; integrates into ETL notebook workflows
Dbt	DQS metrics as Dbt test macros; thresholds in `schema.yml` per model

miss defects surfacing later, potentially undercounting defect events and affecting the correlation estimate. Default weights were derived by expert elicitation; data-driven calibration from labelled defect histories may improve accuracy in other organisations.

6 Conclusion

We presented DQS, a mathematically transparent, reference-based framework for quantifying relational dataset quality in database migration scenarios. DQS evaluates six dimensions against a source reference and aggregates them via risk-aware weighting with formal redistribution for inapplicable dimensions. The harmonic-mean Row Fidelity metric jointly penalises missing and spurious rows without double-counting. Empirical validation on four production healthcare datasets (millions of rows, more than 800 pipeline runs) showed a strong negative correlation ($r < -0.9$, $p < 0.001$) between DQS and downstream data-readiness defects, with a substantial reduction in quality triage time after deployment. Operational experience indicates that composite gating provides greatest value for high-complexity transactional tables with composite PKs, where dimension-specific defects are most likely to interact and evade single-dimension checks.

Future work includes extension to semi-structured migration (JSON, Parquet); data-driven weight calibration; cross-domain validation studies; streaming and CDC migration scenarios; modelling interaction effects among co-occurring defect types; and evaluating DQS as a quality gate for AI/RAG ingestion pipelines.

References

[1] C. Batini and M. Scannapieco. *Data and Information Quality: Dimensions, Principles and Techniques.* Springer, Cham, Switzerland, 2016.

[2] T. C. Redman. *Data Quality: The Field Guide.* Digital Press, Boston, MA, 2001.

[3] E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23(4):3–13, 2000.

[4] Z. Abedjan, L. Golab, and F. Naumann. Profiling relational data: a survey. *VLDB Journal*, 24(4):557–581, 2015.

[5] L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data quality assessment. *Commun. ACM*, 45(4):211–218, 2002.

[6] S. Chug, P. Kaushal, P. Kumaraguru, and T. Sethi. Statistical learning to operationalize a domain agnostic data quality scoring. *arXiv preprint arXiv:2108.08905*, 2021.

[7] F. Bayram, B. S. Ahmed, E. Hallin, and A. Engman. DQSOps: Data quality scoring operations framework for data-driven applications. In *Proc. EASE '23*, ACM, 2023.

[8] ISO/IEC 25012. Software engineering — Data quality model, 2008.

[9] S. Mohammed, L. Ehrlinger, H. Harmouch, F. Naumann, and D. Srivastava. The five facets of data quality assessment. *SIGMOD Record*, 54(2):18–27, 2025.

[10] A. Srivastava and J. K. Multani. Assessing application stability — the Application Stability Index (ASI). *Int. J. Eng. Comput. Sci.*, 15(3):28337–28343, 2026. https://doi.org/10.18535/ijecs/v15i03.5448.

[11] I. Taleb, M. A. Serhani, C. Bouhaddioui, and R. Dssouli. Big data quality framework: a holistic approach to continuous quality management. *J. Big Data*, 8(1):76, 2021.

[12] R. Y. Wang and D. M. Strong. Beyond accuracy: What data quality means to data consumers. *J. MIS*, 12(4):5–33, 1996.

[13] Great Expectations. *Great Expectations: Open Source Data Quality Framework*, v0.18, 2023. https://greatexpectations.io.