

Quality-Based SQL: Specifying Information Quality in Relational Database Queries

Amir Parssian, InfoPyramid

William Yeoh and Mong Shan Ee, Deakin University

Although data is essential to accurate decision making, data errors persist in most enterprise databases, which degrades decision quality. To counteract errors from data-entry mistakes, transaction errors, or system failures¹ and ensure that users get the right data to make key decisions, measurements of information quality are imperative. The saying “if you can’t measure it, you can’t manage it” captures the idea that no one can determine how data quality influences a decision without some measure of data quality.

With that idea in mind, many researchers have conducted studies related to identifying and measuring data-quality characteristics, managing data quality, and determining how data quality affects business operations. Many such efforts have focused on defining data quality and identifying quality dimensions,^{2–4} with the majority of studies concluding that accuracy, completeness, timeliness, and consistency are the most important data-quality dimensions.

Of these, accuracy and completeness are the most widely cited.⁵ The user, or information consumer, is also likely to rate accuracy and completeness the most highly, particularly because other essential data-quality

A Structured Query Language extension uses an estimator module to evaluate quality profiles that rate the accuracy and completeness of query results. Users receive information that matches their defined quality constraints and better serves their data needs.

characteristics, such as timeliness and consistency, are closely tied to these two attributes.⁶ Lack of timeliness could lead to incomplete or inaccurate data, for example, and inconsistency can stem from inaccurate or incomplete data sources.

Data-accuracy and completeness problems often stem from joining data tables in a query search. A relational database management system (RDBMS) that supports marketing analysis for department stores, for example, might access transactions, customer data, and clothing item tables. If an analyst queries the RDMS to return results on “women who live within 40 miles of a major city, have an annual income of at least \$40,000, and have bought products online at least 12 times in the past 6 months,” the query engine might then join the tables to create a list of potential customers for professional work attire. But how complete and accurate is the resulting

FOUNDATIONAL TO QUALITY-BASED SQL IS AN SQL QUALITY ESTIMATOR THAT USES NOVEL METRICS TO EVALUATE QUERY RESULTS AGAINST NUMERICALLY DEFINED QUALITY CONSTRAINTS.

information? The derived data's quality attributes are a function of the respective tables' quality attributes, but the RDBMS is based on the Structured Query Language (SQL), which provides no mechanism for defining the quality of query results. Thus, the analyst has no choice but to accept the returned information at face value.

To address the need for higher-quality query results, we developed Quality-Based SQL (QSQL), an SQL extension that lets users specify the quality level of returned information as part of the query. Foundational to QSQL is an SQL quality estimator that uses novel metrics to evaluate query results against numerically defined quality constraints. Because users see only data that meets or surpasses the set numeric boundary, they are assured of receiving only data that has the desired accuracy and completeness. With that assurance, they can use query results to make more informed business decisions.

DATA ACCURACY AND COMPLETENESS: TWO CASE STUDIES

Although the literature describes many data-quality attributes, accuracy and completeness are the two most widely cited, in part because they can be measured objectively.⁵ Two case studies illustrate the need to specify these attributes and how specifications might vary across business operations.

Retail banking

In retail banking, customer data—demographics, product types, transaction dates, and transaction amounts—is used extensively in billing, marketing, and risk monitoring. Consider a

customer address file that is only 75 percent accurate. If a bank uses the database for billing, it could easily send a statement to the wrong customer, which could have a negative ripple effect. Bank statements typically contain a customer's private information, such as name, account information, and transaction details, so if this error is pervasive, the inaccurate data will contribute to the bank's failure to safeguard its customers' private information, which will undermine the bank's reputation and hence its bottom line.

Data completeness and accuracy are not always absolute, which is another motivation for quantifiable ratings. Inaccurate data hurts the bank in one application, but if the bank's mortgage division uses the database to find potential customers, the 75 percent address accuracy could actually promote a segmented marketing program to identify potential customers. No privacy issues arise if the brochures go to the wrong address, and the recipients might be interested in buying property.

Insurance data

Insurance companies maintain data such as a property's total insured value, location, square footage, and occupancy. Many surveys, such as that by Ernst & Young, show that general insurance and reinsurance industries view this data as central to effective catastrophe risk management.⁷ After the 2004 and 2005 hurricane seasons, Risk Management Solutions reported that the poor quality of data about a property's exposure in a catastrophe (which is essential to modeling loss) contributed to as much as 45 percent of the gap between modeled and actual losses.⁸ According to the report, gap contribu-

tors included missing secondary characteristics such as construction, occupancy, year built, and building height.

In addition to policy information, insurance companies use a property's location and secondary characteristics data to estimate the property's catastrophe risk exposure.⁹ Incomplete data will lead to the underestimation of loss, which could affect the insurance company's solvency if the gap between modeled and actual loss is significant and the company has insufficient capital to meet the actual claims.

The problematic quality of exposure data will also result in higher reinsurance premiums. Insurance companies pay reinsurers a reinsurance premium to pass on the financial obligation for potential catastrophic losses. More than 90 percent of the reinsurers in the Ernst & Young survey acknowledged that they were applying a premium surcharge to compensate for data-quality deficiencies—70 percent of the reinsurers admitted to including up to 25 percent of the premium as a surcharge.

Data incompleteness can also be due to deliberate action, carelessness, or insufficient monitoring at the data entry point. The collapse of Independent Insurance, a UK general insurance company, is a case in point. The company failed to enter a large number of claims into its accounting system and consequently could no longer use any actuarial basis to provide a reasonable estimate of its general liability account in its London market.¹⁰

MEASURING DATA ACCURACY AND COMPLETENESS

Data-quality ratings are generally low, medium, and high, which are inherently subjective—not only are the meanings

vague, but also consumers can disagree about what constitutes high- or low-quality data, as the banking case study illustrates. Quantitative information-quality metrics would promote more objective judgments and decisions.

Measuring data accuracy and completeness within the relational data model has been the subject of recent studies. To execute SQL statements, an RDBMS engine creates a query-execution plan subject to cost optimization that relies on five base-relational algebraic operations—selection, projection, Cartesian product, difference, and union—to translate the SQL commands into relational algebra statements. One research group developed an analytical methodology to derive metrics for measuring data accuracy, completeness, and mismembership for these operations.^{4,5} Other work produced metrics for the accuracy and completeness of five common statistical aggregate functions—COUNT, SUM, MAX, MIN, and AVG.⁶

Our work builds on this foundation by applying the metrics to extend SQL syntax and providing a mechanism for objectively determining the value and usefulness of query results. Obviously, measurements such as “80 percent accurate” are more suitable than “medium accuracy” for this determination.

Within the relational data model, four quality-profile types are possible:⁵

- ▶ **Accurate.** A tuple is accurate if and only if all the attribute values are accurate.
- ▶ **Inaccurate.** A tuple is inaccurate if one or more of its non-key attributes is inaccurate.
- ▶ **Mismember.** A tuple is a mismember if it does not belong to the relation.

TABLE 1. Foundational definitions in Quality-Based SQL.*

Definition	Meaning
S_A	Set of accurate tuples
S_I	Set of inaccurate tuples
S_M	Set of mismember tuples
S_C	Set of incomplete tuples
$ S , S_A , S_I , S_M $ and $ S_C $	Cardinalities of sets S, S_A, S_I, S_M , and S_C
$\alpha_S = S_A / S $	Metric to measure accuracy of S
$\beta_S = S_I / S $	Metric to measure inaccuracy of S
$\mu_S = S_M / S $	Metric to measure mismembership of S
$\chi_S = S_C / (S - S_M + S_C)$	Metric to measure incompleteness of S

* where $0 \leq \alpha_S, \beta_S, \mu_S \leq 1$ and $\alpha_S + \beta_S + \mu_S = 1$

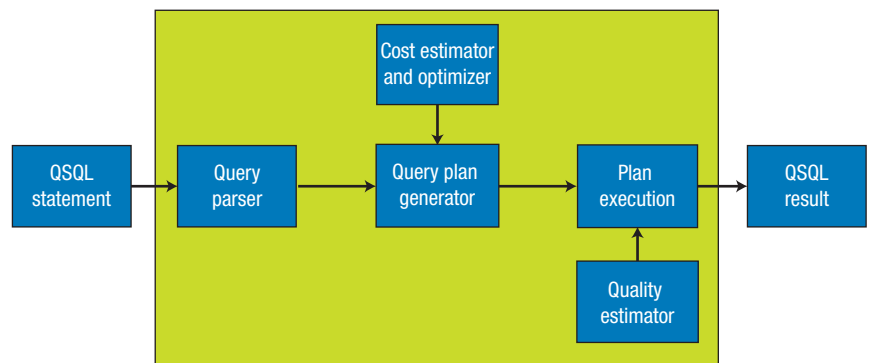


FIGURE 1. Quality-Based SQL (QSQL) architecture. The QSQL statement enters the query processor (green rectangle), which selects an execution plan, executes it, evaluates the information quality, and, if the information meets or exceeds the user’s stated quality needs, presents query results to the user.

- ▶ **Incomplete.** A tuple is incomplete if it belongs to the relation but is not captured into it.

Random database table sampling and auditing will reveal the rate of each tuple type, which the system can then store as metadata. Subsequently, the metrics capture the propagation of these rates from a base table throughout the query result.

The quality metrics to measure accuracy, completeness, and mismembership for the selection, projection, and Cartesian product operations⁵ as well as for a join operation¹¹ are available at <https://goo.gl/oyvbm6>.

Table 1 presents the definitions

underlying the quality metrics for a relation S that contains all tuples captured for a real-world entity type. The definitions serve as QSQL’s theoretical foundation.

QSQL ARCHITECTURE

As Figure 1 shows, our QSQL architecture is an extension of the query-processing modules in mainstream RDBMSs. Users compose their queries and submit them through an interface. The query parser checks for syntax correctness and passes the query to the query plan generator and cost estimator. The optimizer module suggests the optimal execution plan. The plan execution module executes the selected plan and generates the query

TABLE 2. Quality profiles for scenario 1.

Table	Cardinality	Accuracy (%)	Inaccuracy (%)	Mismembership (%)	Completeness (%)
Customer	5,000	80	15	50	85
Order	30,000	90	7	3	95

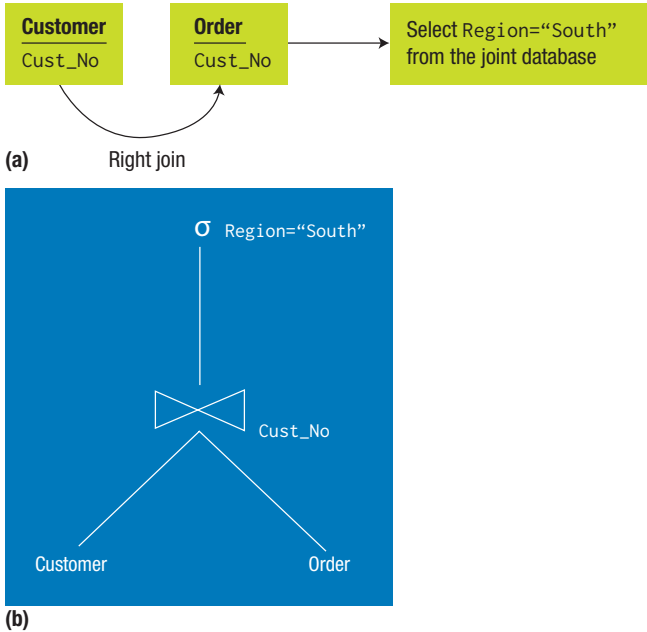


FIGURE 2. Query execution plan (a) as a flowchart and (b) in SQL notation. The system first joins Cust_No from the Customer and Order databases and then obtains tuples that have Region="South".

result. As a last step, the quality estimator evaluates the information quality and, if the quality meets or exceeds the user's defined quality needs, presents the query results to the user.

If there is only a single data source, the quality estimator calculates the query result's quality profile by applying the available metrics. When multiple data sources are available, the quality estimator forces the query processing engine to run the query on all available data sources separately to calculate and compare the quality profiles and provide the query result that best complies with the user's specified information quality needs.

SAMPLE QUERY

A sample query applied in two scenarios illustrates how our proposed architecture works. Each data source in the

scenarios has its own accuracy, inaccuracy, mismembership, and completeness profiles, which the system has determined and recorded as metadata.

The sample query is from a data warehouse architect with access to multiple customer data sources. Her goal is to select a data subset that is 70 or more percent accurate and 85 or more percent complete, and she will load results into the marketing department's data mart. The architect uses the following query to specify and later evaluate the data subset's quality:

```
SELECT*
FROM Customer, Order
WHERE Customer.Cust_No=Order.
      Cust_No
      AND Region="South"
      AND Accuracy ≥ 70%
      AND Completeness ≥ 85%;
```

Scenario 1

In the first query scenario, only one data source is available to the architect. Table 2 shows the quality profiles for this scenario.

The quality profile rates for the base relations can be obtained through conventional sampling schemes. The estimator then audits the sample, which usually contains a few hundred tuples, and determines the quality profiles. Subsequently, the estimator can use statistical inference to obtain quality profiles for all the relations. Figure 2 shows a possible execution plan for scenario 1's query execution

The execution plan has three phases:

- 1. Obtain the Cartesian product of the customer and order tables and store it in a temporary table, say, Temp1.
- 2. Perform selection on Temp1 to retrieve the tuples that have Customer.Cust_No=Order.Cust_No and store the result in another temporary table, say, Temp2. Cust_No is the primary key for the Customer database and also a foreign key in the Order database.
- 3. Perform selection on Temp2 to obtain the tuples that have Region="South". The result will be the query output.

We assume that the query retrieves 2,000 rows. Executing a join operation¹¹ yields the quality profiles in Table 3 for the execution plan and final result.

Interestingly, the query result does not comply with the user's information-quality demands as defined in the QSQL query. Although accuracy is 72 percent—higher than the

TABLE 3. Quality profiles for scenario 1's execution plan and final result.

Table	Cardinality	Accuracy (%)	Inaccuracy (%)	Mismembership (%)	Completeness (%)
Temp1	50 × 10 ⁶	72	20	8	81
Temp2	50,000	72	19	9	80
Query result	2,000	72	18	10	79

user-specified 70 percent—the 79 percent completeness does not meet the user's demand of 85 percent. These profiles imply that the query result does not meet the user's overall information needs, and the quality estimator will return an empty answer to the user, not the 2,000 rows that traditional SQL would present.

Scenario 2

In the second query scenario, the architect can choose from three distinct data sources. To select the source with the highest combination of accuracy and completeness, she would employ this QSQL query:

```
SELECT*
FROM Customer, Order
WHERE Customer.Cust_No=Order.
      Cust_No
AND Region="South"
AND (Accuracy*Completeness) ≥
      0.60;
```

Table 4 gives the possible quality profiles for the three sources.

The query execution plan is applied to each source separately and the quality profiles of the query results are estimated, as summarized in Table 5.

The product of accuracy and completeness for source 1 is 0.64 (0.72 × 0.89), 0.37 for source 2, and 0.55 for source 3. Clearly, source 1 complies with all the query constraints and will be chosen.

QUALITY PROFILE APPLICATIONS

The QSQL concept will help information consumers determine if the derived information meets the desired quality levels for the intended task, and whether data cleaning is needed. Several

TABLE 4. Quality profiles for the three data sources in scenario 2.

Source	Table	Cardinality	Accuracy (%)	Inaccuracy (%)	Mismembership (%)	Completeness (%)
1	Customer	5,000	80	10	10	95
	Order	30,000	90	5	5	95
2	Customer	7,000	70	20	10	70
	Order	50,000	85	5	10	75
3	Customer	3,000	90	5	5	80
	Order	20,000	90	5	5	85

TABLE 5. Quality profiles for the query results in scenario 2.

Source	Table	Cardinality	Accuracy (%)	Inaccuracy (%)	Mismembership (%)	Completeness (%)
1	Query result	2,000	72	12	16	89
2	Query result	5,000	60	19	21	61
3	Query result	1,000	81	8	12	67

applications illustrate the potential benefits of metrics-based quality profiles.

Medical diagnosis

The generation of quality profiles in a hospital's electronic health record (EHR) system provides medical staff with sufficiently reliable information to make an informed clinical diagnosis. The staff member uses the selection and projection operations to extract patient medical history and medication records. The quality profile ensures that the medication records have an acceptable percentage of missing values. Without the profile, the staff member would not recognize that the information being extracted from the EHR is incomplete.

Financial risk analysis

Generating quality profiles on several databases in a bank's customer information file (CIF) gives the risk analyst sufficient assurance of information quality to accurately estimate a debt portfolio's credit risk exposure. The analyst first uses the loan's credit rating to estimate loss exposure and then applies the selection and join operations to extract loss exposure according to debt type. If the results of the quality profiles do not meet the company's data-quality benchmarks, the risk analyst can cross-check the results with the credit rating supplied by other credit rating agencies to verify that the credit rating assigned to the debt instruments is correct. Without the quality profiles,

ABOUT THE AUTHORS

AMIR PARSSIAN is the chief systems architect at InfoPyramid, based in Michigan. His research interests include data quality, data mining, and data warehousing. Parssian received a PhD in management information systems from the University of Texas at Dallas. He is a member of the Association for Information Systems (AIS) and ACM. Contact him at parssian@gmail.com.

WILLIAM YEOH is codirector of the IBM Centre of Excellence in Business Analytics at Deakin University in Melbourne, Australia. His research interests include business intelligence and analytics, information quality, and information systems. Yeoh received a PhD in business intelligence from the University of South Australia. He is a member of AIS. Contact him at william.yeoh@deakin.edu.au.

MONG SHAN Ee is a lecturer in the Department of Finance at Deakin University. Her research interests include pricing optimization, data mining, and optimal stopping problems. Ee received a PhD in engineering from the University of Tsukuba, Japan. Contact her at mong.e@deakin.edu.au.


the risk analyst might not be aware that the information extracted from the CIF is inaccurate and thus will fail to cross-check it with the rating.

Data-quality analysis

Using quality profiles enables data analysts to better control and maintain the quality of data warehouses. With the profiles, an analyst can determine whether data quality meets acceptable tolerance levels for the company's key data elements. If it does not, the analyst can take steps, such as filtering, correction, and verification, to fix the identified data-quality problems. The quality profiles also enable the analyst to measure the progress toward achieving or even surpassing tolerance levels.

Our goal in creating QSQL was to extend SQL to better support decision makers. Knowledge about the quality of query results enables managers to support, modify, or even cancel their business decisions. More informed decisions save overhead and spillover costs and increase an organization's profitability.

To our knowledge, QSQL is the first attempt to incorporate user-specified

quality constraints into SQL. We hope that our work to extend existing relational database and query-processing engines through a metrics-based quality estimator module will serve as the basis for future studies on quality-based SQL. 

REFERENCES

1. B.D. Klein, D.L. Goodhue, and G.B. Davis, "Can Humans Detect Errors in Data? Impact of Base Rates, Incentives, and Goals," *MIS Q.*, vol. 21, no. 2, 1997, pp. 169–194.
2. Y. Wand and R.Y. Wang, "Anchoring Data Quality Dimensions in Ontological Foundations," *Comm. ACM*, vol. 39, no. 11, 1996, pp. 86–95.
3. M. Jarke et al., "Architecture and Quality in Data Warehouses: An Extended Repository Approach," *Information Systems*, vol. 24, no. 3, 1999, pp. 229–253.
4. A. Parssian, S. Sarkar, and V.S. Jacob, "Impact of the Union and Difference Operations on the Quality of Information Products," *Information Systems Research*, vol. 20, no. 1, 2009, pp. 99–120.
5. A. Parssian, S. Sarkar, and V.S. Jacob, "Assessing Data Quality for Information Products: Impact of Selection, Projection, and Cartesian Product,"

Management Science, vol. 50, no. 7, 2004, pp. 967–982.

6. A. Parssian, "Managerial Decision Support with Knowledge of Accuracy and Completeness of Relational Aggregate Functions," *Decision Support Systems*, vol. 42, no. 3, 2006, pp. 1494–1502.
7. Ernst & Young, "Raising the Bar on Catastrophe Data: The Ernst & Young 2008 Catastrophe Exposure Data Quality Survey," 2013; www.acordlondon.org/Documents/Ernst_Young_Catastrophe_Exposure_Data_Quality_Survey.pdf.
8. A. Lavakare, "C-Level Agenda: Exposure Data Quality Is a Key Indicator of Operating Risk," *Best's Rev.*, 1 Dec. 2008, pp. 82–84.
9. P. Grossi, H. Kunreuther, and D. Windeler, "An Introduction to Catastrophe Models and Insurance," *Catastrophe Modeling: A New Approach to Managing Risk*, P. Grossi and H. Kunreuther, eds., Springer Science + Business Media, 2005, pp. 23–42.
10. "Independent Goes into Liquidation," *BBC News*, 18 June 2001; <http://news.bbc.co.uk/2/hi/business/1394424.stm>.
11. A. Parssian, S. Sarkar, and V.S. Jacob, "Assessing Information Quality for the Composite Relational Operation Join," *Proc. 7th Int'l Conf. Information Quality (ICIQ 02)*, 2002, pp. 225–237.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.