Assessing Information Quality for the Composite Relational Operation Join

Amir Parssian
College of Business and Management
University of Illinois at Springfield
One University Plaza
Springfield, Illinois 62703-5407
apars1@uis.edu

Sumit Sarkar and Varghese S. Jacob
School of Management
University of Texas at Dallas
Richardson, Texas, 75080
sumit ; vjacob@utd.edu

Abstract: Information plays an increasingly important role in strategic decision-making processes within businesses. Therefore, information quality and its assessment have become critical subjects for information products delivered to information consumers. Commonly, the information product provided to consumers is the output of queries in relational databases. The queries typically consist of one or more primitive relational algebra operations. Previous research has addressed the measurement of important quality attributes of the output of primitive relational algebra operations such as selection, projection, and Cartesian product. In this paper, we present a methodology to measure the quality profile of the output of the relational operation join, that is one of the most widely used composite operation. Different types of join operations are identified based on the attributes that participate in the join condition and the output quality profile for each of these types of the join operation is derived. Examples are provided to highlight the differences between the quality profile of the input relations and those of the output of the join operation.

Keywords: Data Quality, Information Quality, Quality Metrics, Relational Algebra

1. Introduction

Businesses are increasingly using their enterprise data for their strategic decision-making activities. In fact, information (derived data) has become one of the most important tools for businesses to gain competitive advantage. Due to the increased importance of data and information, their quality assessment has also come under considerable attention in both academic and practitioner circles. Substantial research has been conducted to identify, define, and characterize the dimensions of data quality [4,6,7]. Business impacts of data quality have also been addressed, and quality issues in data management processes have been identified as a critical issue [1].

In order to examine the impact of the quality of information on the quality of a decision, the information quality needs to first be measured. Among many data quality dimensions studied and reported in the literature, we focus on metrics associated with two quality attributes, accuracy and incompleteness, that are of critical importance to information consumers. Many of the other data quality dimensions are closely tied to these two. For instance, the lack of timeliness leads to incompleteness or inaccuracy of the data available to end-users. Similarly, data inconsistency is usually caused by inaccuracies in the data or incompleteness of the data.

Given the widespread use of the relational data model in practice, we examine quality assessment for information products for relational databases. The quality dimensions can be measured at various levels of granularity, e.g., cells, tuples, attributes, or relations. We focus on quality assessments at the relation level for two important reasons. First, users are often provided information in a tabular form. Second, the more detailed the granularity, the more expensive it is to measure and represent the quality metrics [5].

In a relational environment, the information product delivered to end-users is usually the output of a query that is typically derived from one or more relations. In previous research, we have developed metrics for the output of the primitive relational algebra operations *selection*, *projection*, and *Cartesian product* [3]. In this research, we present a methodology to assess the quality profiles for the output of queries that include more than one of these primitive operations. Specifically, we focus on the relational operation *join* since it is very widely used in querying databases. Different types of join operations are identified based on the attributes that participate in the join condition, and quality profiles of the output for each such type are derived. There has been some related prior research. Kon et al. [2] presented an error representation schema consisting of three error types namely, inaccuracy, incompleteness, and mismembership, and showed the closure property of these error types under the relational algebra operations. They did not, however, provide a methodology to operationalize their framework. Reddy and Wang [5] provided an analysis of the error propagation process when only inaccuracies and mismembers are important. In this work, we draw upon the prior research where appropriate in order to address the quality metrics for the output of the different types of the join operation.

The rest of this paper is organized as follows. In Section 2, we discuss the error types and their base metrics. The metrics for the results of the three primitive operations (selection, projection, and Cartesian product) are summarized in Section 3. The quality metrics for the different types of join operations are discussed in section 4. We illustrate our work with a numerical example in Section 5, and provide our concluding notes in Section 6.

2. Error Types and Base Metrics

2.1 Errors Types

To provide a formal definition of the error types that we are interested in, consider the notion of a conceptual relation, denoted by T, which represents the underlying instances and their attributes of interest for a true world entity (e.g., potential customers). A business may store the data on such entity instances in a relation S. The relationship between T and S, shown in Figure 2.1, helps in identifying the nature of errors and the factors that lead to those errors.

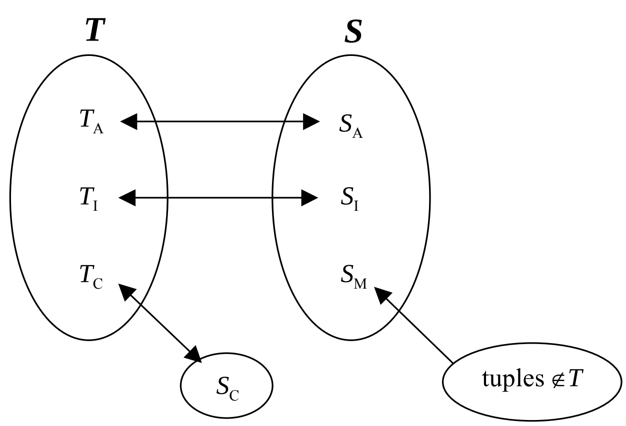

Fig. 2.1 Mapping of the data sets of S and T

In an ideal world, all the relevant attributes of each entity instance in T would be correctly captured in S. That is usually not the case in practice. Some of the entity instances in the real world captured by T_A, are represented correctly in S (denoted by S_A). For some others, T_I, only part of the attributes may be correctly represented (S_I). Some entity instances in the real world, T_C ($\cong S_C$), may not appear in S at all. A few instances, S_M , which are stored in S may not correspond to any entity instance in the real world of interest.

Let t_i refer to an entity instance in T and s_j refer to an entity instance stored in S. t_{ik} (s_{jk}) refers to the k^{th} attribute value for t_i (s_j). Let n be the total number of attributes of interest. Further, assume that the set of attributes indexed by $1,...,m$ refer to the set of identifier attributes and the set indexed by $m+1,...,n$ refer to non-identifier attributes. We can then state the following:

- A tuple $s_j \in S$ is **Accurate** *iff* $\{\exists\, t_i \in T \mid (s_{jk} = t_{ik})\ \forall\, k=1,\ldots,m \wedge (s_{jl} = t_{il}),\ l=m+1,..,n\}$;
- A tuple $s_j \in S$ is **Inaccurate** *iff* $\{\exists\, t_i \in T \mid (s_{jk} = t_{ik})\ \forall\, k=1,\ldots,m \wedge \exists\, (s_{jk} \neq t_{il}),\ l=m+1,..,n\}$;
- A tuple $s_j \in S$ is a **Mismember** *iff* $\{\neg\exists\, t_i \in T \mid (s_{jk} = t_{ik})\ \forall\, k=1,\ldots,m\}$;
- An instance $t_i \in T$ belongs to the **Incomplete** set S_C *iff* $\{(\neg\exists\, s_j \in S) \mid (s_{jk} = t_{ik})\ \forall\, k=1,\ldots,m\}$.

It is worth noting that inaccurate values in the identifier attributes lead to mismembership, since the stored data refer to entity instances that do not belong to the relevant real world. The above definitions are analogous to those provided by Kon et al. [2]. Reddy and Wang [5] provided similar (but not identical) definitions for inaccuracy and mismembership, and did not consider incompleteness. Interested readers are referred to the cited articles for a comprehensive discussion on how the different errors appear in the data.

2.2 Base Metrics

We describe metrics for the errors using the definitions for accuracy and for the different error types.

Accuracy of *S*, denoted by α_S, is defined as the proportion of tuples in S that are accurate, i.e., $\alpha_S = \frac{|S_A|}{|S|}$, where $|S|$ and $|S_A|$ are the cardinalities of S and its accurate subset S_A, respectively.

Inaccuracy of *S*, denoted by β_S, is defined as the proportion of tuples in S that are inaccurate, i.e., $\beta_S = \frac{|S_I|}{|S|}$, where $|S_I|$ is the cardinality of the inaccurate subset S_I. This metric for inaccuracy differs from that of Reddy and Wang [5] as it does not include mismembers caused by incorrectly stored values of one or more identifier attributes. We prefer this interpretation because entity instances are identified by their identifier attribute values, and such an error identifies incorrect entity instances in the relevant real world.

Mismembership of *S*, denoted by μ_S, is defined as the proportion of tuples in S that do not correspond to entity instances in the real world, i.e., $\mu_S = \frac{|S_M|}{|S|}$, where $|S_M|$ is the cardinality of the mismember subset S_M.

Incompleteness of *S*, denoted by χ_S, is defined as the proportion of entity instances in the relevant real world (T) that is not represented in S, i.e., $\chi_S = \frac{|T_C|}{|T|} = \frac{|S_C|}{|S| - |S_M| + |S_C|}$, where $|T|$ is the cardinality of the set T, and $|T_C|$ ($= |S_C|$) is the cardinality of the subset T_C.

2.3 Estimation Issues

To illustrate how the base metrics are estimated, we consider a real world entity type *Customer*. Sample data for the conceptual (T), stored (S), and the incomplete data set (S_C) are shown in Figures 2.2 and 2.3., respectively.

T

Cust_ID	Cust_Name	City
C1	Boeing	Los Angeles
C2	Coca Cola	Atlanta
C3	Chrysler	Los Angeles
C5	Microsoft	Seattle

S

Cust_ID	Cust_Name	City	Tuple Status
C1	Boeing	Los Angeles	A
C2	Coca Cola	Atlanta	A
C3	Chrysler	New York	I
C4	IBM	New York	M

Fig. 2.2 Conceptual (T) and stored (S) relations for the real world entity customer

$\boldsymbol{S}_{\mathrm{C}}$

Cust_ID	Cust_Name	City
C5	Microsoft	Seattle

Fig. 2.3 Incomplete dataset for the customer entity

Cells with inaccurate values are shown with a black background, and mismember tuples are shown with a gray background. The tuple status column in relation *S* (Fig. 2.2) indicates whether a tuple is accurate (A), inaccurate (I), or a mismember (M). Note that the tuple status column is shown for illustrative purposes and is not actually stored in *S*. Data about the customer identified by Cust_ID= 'C5' has not been captured in *S* as it should have and therefore it forms the incomplete data set for *S*.

We need the parameters $|S|$, $|S_A|$, $|S_I|$, $|S_M|$, and $|S_C|$ to determine the base metrics for *S*. In practice, it is usually not possible to verify all tuples in *S* in order to determine these parameters. Instead, sampling techniques can be used to assess these parameters. Estimating α_S, β_S, and μ_S are generally straightforward. In order to estimate χ_S, it is necessary to obtain a sample of the real world entity instances, and then verify what proportion is represented in the database.

3. Metrics for the Primitive Operations

We provide in summary the results of our analysis for selection, projection, and the Cartesian product as they are the primitive operations that constitute the various types of join operations. Details of this analysis have been presented in [3].

3.1 Selection

We denote by *R* the result obtained by applying the selection operation and distinguish between the following cases for this operation:

1) The selection condition applies to an identifier attribute of *S*;
2) The selection condition applies to a non-identifier attribute of *S*;
   a) The selection condition is an inequality (i.e., contains '<' or '>'); and
   b) The selection condition is an equality (i.e., contains =').

We have developed metrics for the various selection scenarios. For instance, when the selection condition applies to an identifier attribute of *S*, the quality profiles of *R* are identical to those of *S*. This is because the status of all selected tuples remains unchanged. In cases where the selection condition applies to a non-identifier attributes of *S* and contains the operator '=' (i.e., case 2.b), the quality profiles for *R* are obtained as [3]:

i) $$\alpha_R = \alpha_S \cdot \frac{|S|}{|R|} \cdot \frac{1-\sqrt{1-4\cdot(1-\gamma_S)\cdot\frac{|R|}{|S|}}}{2\cdot(1-\gamma_S)};$$

ii) $$\beta_R = \left((\alpha_S+\beta_S)\cdot\gamma_S - \alpha_S\right)\cdot \frac{|S|}{|R|} \cdot \frac{1-\sqrt{1-4\cdot(1-\gamma_S)\cdot\frac{|R|}{|S|}}}{2\cdot(1-\gamma_S)};$$

iii) $$\mu_R = 1-\left((1-\mu_S)\cdot\gamma_S\cdot\frac{|S|}{|R|}\cdot\frac{1-\sqrt{1-4\cdot(1-\gamma_S)\cdot\frac{|R|}{|S|}}}{2\cdot(1-\gamma_S)}\right);\text{ and}$$

iv) $$\chi_R = 1-(1-\chi_S)\cdot\gamma_S.$$

where $\gamma_S = \left(\dfrac{\alpha_S}{\alpha_S + \beta_S} \right)^{\frac{1}{q_S}}$ is the non-identifier attribute accuracy and q_S is the number of non-identifier attributes in *S*. Similarly, quality profiles for case 2.a have been obtained [3]. It is worth noting here that the quality profiles for case 2.a and 2.b are different.

3.2 Projection

For the projection operation, an important consideration is the normalization scheme of the base relation *S* since it affects the formation of the set of identifier attributes for *R*. Knowing the identifier attributes for *R* is essential for categorization of tuples in the output of the projection operation. We have developed metrics for the general projection scenario where a subset of identifier attributes along with a subset of non-identifier attributes of *S* are projected into *R* [3]. Other projection scenarios such as when only a subset of identifier attributes of *S* is projected into *R* are handled as special cases of the general scenario.

3.3 Cartesian Product

When evaluating the quality profiles for the result of the Cartesian product operation, it is necessary to first be able to categorize the resulting tuples. We have established the tuple categorization scenarios for the Cartesian product operation applied to two base relations S_1 and S_2 [3]. Let α_1, β_1, μ_1, and χ_1 indicate the quality profiles of S_1, and α_2, β_2, μ_2, and χ_2 indicate the quality profiles of S_2. The quality profiles for *R* are given by:

i) $\alpha_R = \alpha_1 \cdot \alpha_2$;

ii) $\beta_R = \alpha_1 \cdot \beta_2 + \alpha_2 \cdot \beta_1 + \beta_1 \cdot \beta_2$;

iii) $\mu_R = \mu_1 + \mu_2 - \mu_1 \cdot \mu_2$; and

iv) $\chi_R = \chi_1 + \chi_2 - \chi_1 \cdot \chi_2$.

4. Quality Metrics for the Join Operation

4.1 Basic Definitions

Two variations of the join operation that are commonly used in queries are the *θ–join* and the *natural join*. We briefly describe them below.

θ–Join: This operation, denoted by $R = S_1 \bowtie_\theta S_2$, returns a relation containing all possible tuples that are a concatenation of two tuples, one from each of two specified relations (denoted by S_1 and S_2), such that the two tuples contributing to any given combination are compared on a common attribute and on the basis of some arithmetic comparison operator (=, < , > , etc.). If θ is '=', then the θ-Join is called an equi-join.

Natural Join: This operation, denoted by $R = S_1 \bowtie S_2$, is an equi-join where the common attributes appear just once, not twice, in the resulting relation.

The natural join, though distinct from the θ–Join, is easily analyzed based on the analysis for the θ–Join. For this reason, we analyze the quality profiles for the output of the θ–Join first, and subsequently extend the analysis to the output of a natural join.

4.2 Quality Metrics for the θ–Join Operation

An important consideration for analyzing the quality profile for the θ–Join is whether the attributes that participate in the join condition (hereinafter referred to as the *conditioning attributes*) are part of the identifier for the corresponding relations. This is because the categorization of a tuple in the result (as accurate, inaccurate, mismember, or incomplete) is determined by the inaccuracies that may be present in

the conditioning attributes. Consequently, this affects the quality profile of the output. We identify the following scenarios that lead to different quality profiles.
1) The join condition applies to attributes both of which are part of the identifier in the corresponding relations.
2) The join condition applies to attributes neither of which are part of the identifier in the corresponding relations.
3) The join condition applies to an attribute that is part of the identifier in one participating base relation, and an attribute that is not part of the identifier in the other relation.
We illustrate these cases with examples and provide the methodology to derive the associated quality profiles. Before doing that, we discuss how the θ–Join operation can be decomposed into the primitive operations selection and Cartesian product, as this phenomenon is common across all of the three scenarios. The θ–Join operation is a composite operation that can be decomposed as a combination of a Cartesian product operation followed by a selection operation, where the selection condition captures the join condition. For expositional purposes, the θ–Join can be modeled as a two-stage process. In the first stage, the Cartesian product of two base relations S_1 and S_2 is obtained and stored in a temporary table denoted by S_{temp}, i.e., $S_{temp} = S_1 \times S_2$. Note that the combination of the identifier (non-identifier) attributes of S_1 and S_2 forms the identifier (non-identifier) attributes for S_{temp}. In the second stage, the selection operation (with appropriate join condition) is applied to S_{temp} to provide the desired result R, i.e., $R = \sigma_\theta(S_{temp})$. The above three join scenarios correspond to the following types of selection conditions.
1) The selection condition applies to attributes that are part of the identifier of S_{temp};
2) The selection condition applies to attributes none of which are part of the identifier of S_{temp};
3) The selection condition applies to attributes one of which is part of the identifier of S_{temp}, and the other that is not part of the identifier of S_{temp}.
In all of these three scenarios, the quality profiles for S_{temp} can be obtained using the results derived for the Cartesian product operation. Let α_{temp}, β_{temp}, μ_{temp}, and χ_{temp} indicate the quality profiles for S_{temp}. The quality profiles for S_{temp} are then obtained as:

$$\alpha_{temp} = \alpha_1 \cdot \alpha_2 \tag{4.1}$$

$$\beta_{temp} = \alpha_1 \cdot \beta_2 + \alpha_2 \cdot \beta_1 + \beta_1 \cdot \beta_2 \tag{4.2}$$

$$\mu_{temp} = \mu_1 + \mu_2 - \mu_1 \cdot \mu_2 \tag{4.3}$$

$$\chi_{temp} = \chi_1 + \chi_2 - \chi_1 \cdot \chi_2 \tag{4.4}$$

We subsequently use these expressions for all the three scenarios.
Before discussing the three scenarios in detail, we illustrate the θ–*Join* for scenario 2 with an example. Consider two base relations S_1 (Customers) and S_2 (Products) as shown in Figure 4.1. The identifying attributes for these relations are Cust_ID and Prod_ID, respectively.

S_1

Cust_ID	Cust_Name	City	Tuple Status
C1	Boeing	Los Angeles	A
C2	Coca Cola	Atlanta	A
C3	Chrysler	New York	I
C4	IBM	New York	M

S_2

Prod_ID	Prod_Desc	Weight	City	Tuple Status
P1	Bolt	12	New York	A
P2	Screw	15	Los Angeles	I
P3	Nut	14	Denver	I
P4	Cog	11	Los Angeles	A
P5	Foam	12	Seattle	M

Fig. 4.1 Stored relations for customers and products

Consider the join condition S_1·City = S_2·City. First, the Cartesian product of S_1 and S_2 can be obtained and stored in S_{temp} as shown in Figure 4.2. The status of tuples in Figure 4.2 are obtained according to the Cartesian product tuple categorization [3]. Next, we obtain $R = \sigma_{S_1.City= S_2.City}$ (S_{temp}) as shown in Figure 4.3. The incomplete data set R_C is shown in Figure 4.4.

4.2.1 Scenario 1: The join condition applies to identifier attributes of participating relations

This corresponds to selection case 1 (section 3.1), and therefore the quality profiles for *R* are identical to those of S_{temp} (expressions 4.1-4.4), i.e., $\alpha_R = \alpha_{temp}$; $\beta_R = \beta_{temp}$; $\mu_R = \mu_{temp}$; and $\chi_R = \chi_{temp}$.

Note that this result applies regardless of whether the join condition applies to the entire identifier attributes of S_1 and S_2, or, to a subset of the identifier attributes of either S_1 or S_2 (or both).

S_{temp}

Cust_ID	Cust_Name	S_1.City	Prod_ID	Prod_Desc	Weight	S_2.City	Tuple Status
C1	Boeing	Los Angeles	P1	Bolt	12	New York	A
C1	Boeing	Los Angeles	P2	Screw	15	Los Angeles	I
C1	Boeing	Los Angeles	P3	Nut	14	Denver	I
C1	Boeing	Los Angeles	P4	Cog	11	Los Angeles	A
C1	Boeing	Los Angeles	P5	Foam	12	Seattle	M
C2	Coca Cola	Atlanta	P1	Bolt	12	New York	A
C2	Coca Cola	Atlanta	P2	Screw	15	Los Angeles	I
C2	Coca Cola	Atlanta	P3	Nut	14	Denver	I
C2	Coca Cola	Atlanta	P4	Cog	11	Los Angeles	A
C2	Coca Cola	Atlanta	P5	Foam	12	Seattle	M
C3	Chrysler	New York	P1	Bolt	12	New York	I
C3	Chrysler	New York	P2	Screw	15	Los Angeles	I
C3	Chrysler	New York	P3	Nut	14	Denver	I
C3	Chrysler	New York	P4	Cog	11	Los Angeles	I
C3	Chrysler	New York	P5	Foam	12	Seattle	M
C4	IBM	New York	P1	Bolt	12	New York	M
C4	IBM	New York	P2	Screw	15	Los Angeles	M
C4	IBM	New York	P3	Nut	14	Denver	M
C4	IBM	New York	P4	Cog	11	Los Angeles	M
C4	IBM	New York	P5	Foam	12	Seattle	M

Fig. 4.2 Cartesian product of S_1 and S_2

R

Cust_ID	Cust_Name	S_1.City	Prod_ID	Prod_Desc	Weight	S_2.City	Tuple Status
C1	Boeing	Los Angeles	P2	Screw	15	Los Angeles	I
C1	Boeing	Los Angeles	P4	Cog	11	Los Angeles	A
C3	Chrysler	New York	P1	Bolt	12	New York	M
C4	IBM	New York	P1	Bolt	12	New York	M

Fig. 4.3 Customers and products with equal value for attribute City

$\boldsymbol{R}_C$

Cust_ID	**Cust_Name**	**S_1.City**	**Prod_ID**	**Prod_Desc**	**Weight**	**S_2.City**
C3	Chrysler	Los Angeles	P2	Screw	16	Los Angeles

Fig. 4.4 Incomplete data set for *R*

4.2.2 Scenario 2: The join condition applies to non-identifier attributes of participating relations

The set of non-identifier attributes of S_{temp} is composed of all non-identifier attributes of S_1 and of S_2, respectively. Therefore, the inaccurate tuples in S_{temp} are those tuples that have at least one inaccurate value for the non-identifier attributes of S_1 or S_2. An important consideration here is that the status of inaccurate tuples in S_{temp} might change when selected into *R*. This is analogous to the result of the selection operation that has been discussed in prior research [Parssian et al., 2002]. For instance, the tuple identified by ('C1','P2') in S_{temp} (Fig. 4.2) satisfies the Join condition and therefore is selected into *R*. The categorization of this tuple as an inaccurate in *R* is due to the inaccuracy of a non-identifier attribute (i.e., Weight) other than the conditioned attribute. The tuple identified by ('C3','P1') satisfies the join condition due to the inaccuracy of one of the conditioned attributes (i.e., S_1·City which should have been recorded as 'Los Angeles'). If the correct value for this attribute were recorded in S_1, then the corresponding tuple in S_{temp} would have not been selected into *R*. Therefore, this tuple is categorized as a mismember in *R*. The tuple identified by ('C3','P2') does not satisfy the join condition and therefore is not selected into *R*. If the correct value for the conditioned attribute were recorded (i.e., if S_1·City was recorded as 'Los Angeles'), then the corresponding tuple in S_{temp} would have been selected into *R*. Therefore this tuple becomes part of the incomplete set R_C.

The quality profiles for *R* can be obtained by using the results of the applicable selection scenario. In this instance, the selection condition contains the '=' operator (selection case 2.b), and therefore we have:

$$|S_{temp}| = |S_1| \cdot |S_2| \tag{4.5}$$

$$\gamma_{temp} = \left(\frac{\alpha_1}{\alpha_1 + \beta_1}\right)^{\frac{1}{q_1}} \cdot \left(\frac{\alpha_2}{\alpha_2 + \beta_2}\right)^{\frac{1}{q_2}} \tag{4.6}$$

$$\alpha_R = \alpha_{temp} \cdot \frac{|S_{temp}|}{|R|} \cdot \frac{1 - \sqrt{1 - 4 \cdot \frac{|R|}{|S_{temp}|} \cdot (1 - \gamma_{temp})}}{2 \cdot (1 - \gamma_{temp})}; \tag{4.7}$$

$$\beta_R = ((\alpha_{temp} + \beta_{temp}) \cdot \gamma_{temp} - \alpha_{temp}) \cdot \frac{|S_{temp}|}{|R|} \cdot \frac{1 - \sqrt{1 - 4 \cdot \frac{|R|}{|S_{temp}|} \cdot (1 - \gamma_{temp})}}{2 \cdot (1 - \gamma_{temp})} \tag{4.8}$$

$$\mu_{\mathrm{R}} = 1 - \left((1 - \mu_{\mathrm{temp}}) \cdot \gamma_{\mathrm{temp}} \cdot \frac{|S_{\mathrm{temp}}|}{|R|} \cdot \frac{1 - \sqrt{1 - 4 \cdot \frac{|R|}{|S_{\mathrm{temp}}|} \cdot \left(1 - \gamma_{\mathrm{temp}}\right)}}{2 \cdot \left(1 - \gamma_{\mathrm{temp}}\right)} \right) \tag{4.9}$$

$$\chi_{\mathrm{R}} = 1 - (1 - \chi_{\mathrm{temp}}) \cdot \gamma_{\mathrm{temp}} \tag{4.10}$$

Note that in (4.6), q_1 and q_2 denote the number of non-identifier attributes in S_1 and S_2, respectively. Substituting for α_{temp}, β_{temp}, μ_{temp}, and χ_{temp} (from equations 4.1-4.4) in the expressions above we obtain the final expressions for the quality profile of *R*.

4.2.3 Scenario 3: The join condition applies to an identifier attribute of one participating relation, and a non-identifier attribute of the other relation

We illustrate this case by an example where S_1 and S_2 are as shown in Figure 4.5. The identifying attributes for these relations are Cust_ID and Order_No, respectively.

S_1

Cust_ID	Cust_Name	City	Tuple Status
C1	Boeing	Los Angeles	A
C2	Coca Cola	Atlanta	A
C3	Chrysler	New York	I
C4	IBM	New York	M
C5	Dell	Dallas	M

S_2

Order_No	Cust_ID	AMT	Tuple Status
O1	C1	100	A
O2	C1	200	I
O3	C2	300	I
O4	C3	400	A
O5	C2	500	M
O6	C3	300	M
O7	C4	100	A
O8	C4	400	M
O9	C5	500	I
O10	C4	200	I
O11	C3	300	I
O12	C3	500	I

Fig. 4.5 Stored relations for customers and orders

For this example, consider the equi-join condition S_1·Cust_ID = S_2·Cust_ID. The intermediate result S_{temp} (the Cartesian product of S_1 and S_2) is shown in Figure 4.6. The join condition in this case applies to a subset of the identifier attributes (i.e., S_1·Cust_ID) and a subset of the non-identifier attributes (i.e., S_2·Cust_ID) of S_{temp}. The result $R = \sigma_{S_1.\mathrm{Cust\_ID} = S_2.\mathrm{Cust\_ID}}\,(S_{\mathrm{temp}})$ is shown in Figure 4.7. The incomplete dataset R_C is shown in Figure 4.8.

Note the change in status of tuples in S_{temp} and *R*. To discuss the tuple status in *R*, let t_1 be a tuple in S_1, t_2 be a tuple in S_2, t_{temp} be a tuple in S_{temp}, and *t* be a tuple in *R* (R_C). We denote the set of inaccurate tuples in S_2 that have an accurate (inaccurate) value for one of the conditioned attribute by $\hat{S}_{2\mathrm{I}}$ ($\tilde{S}_{2\mathrm{I}}$). Then, we recognize the following categorizations for tuples in *R* as summarized in Figure 4.9.

S_{temp}

S_1.Cust_ID	Cust_Name	City	Order_No	S_2.Cust_ID	AMT	Tuple Status
C1	Boeing	Los Angeles	O1	C1	100	A
C1	Boeing	Los Angeles	O2	C1	200	I
C2	Coca Cola	Atlanta	O3	C2	300	I
C2	Coca Cola	Atlanta	O5	C2	500	M
C3	Chrysler	New York	O4	C3	400	I
C3	Chrysler	New York	O6	C3	300	M
C3	Chrysler	New York	O11	C3	300	I
C3	Chrysler	New York	O12	C3	500	I
C4	IBM	New York	O7	C4	100	M
C4	IBM	New York	O8	C4	400	M
C4	IBM	New York	O10	C4	200	M
C5	Dell	Dallas	O9	C5	500	M
…	…	…	…	…	…	…
…	…	…	…	…	…	…

Fig. 4.6 Cartesian product of S_1 and S_2

R

S_1.Cust_ID	Cust_Name	City	Order_No	S_2.Cust_ID	AMT	Tuple Status
C1	Boeing	Los Angeles	O1	C1	100	A
C1	Boeing	Los Angeles	O2	C1	200	I
C2	Coca Cola	Atlanta	O3	C2	300	M
C2	Coca Cola	Atlanta	O5	C2	500	M
C3	Chrysler	New York	O4	C3	400	M
C3	Chrysler	New York	O6	C3	300	M
C3	Chrysler	New York	O11	C3	300	M
C3	Chrysler	New York	O12	C3	500	M
C4	IBM	New York	O7	C4	100	M
C4	IBM	New York	O8	C4	400	M
C4	IBM	New York	O10	C4	200	M
C5	Dell	Dallas	O9	C5	500	M

Fig. 4.7 Query result for the join case 3

R_C

S_1.Cust_ID	Cust_Name	City	Order_No	S_2.Cust_ID	AMT
C1	Boeing	Los Angeles	O12	C1	500

Fig. 4.8 Incomplete dataset for the query result

	$t_2 \in S_{2A}$	$t_2 \in \hat{S}_{2I}$	$t_2 \in \widetilde{S}_{2I}$	$t_2 \in S_{2M}$
$t_1 \in S_{1A}$	$t \in R_A$ (‘C1’,’O1’)	$t \in R_I$ (‘C1’,’O2’)	$t \in R_M$ $t \in R_C$ (‘C2’,’O3’)	$t \in R_M$ (‘C2’,’O5’)
$t_1 \in S_{1I}$	$t \in R_I$ (‘C3’,’O4’)	$t \in R_I$ (‘C3’,’O11’)	$t \in R_M$ $t \in R_C$ (‘C3’,’O12’)	$t \in R_M$ (‘C3’,’O6’)
$t_1 \in S_{1M}$	$t \in R_M$ (‘C4’,’O7’)	$t \in R_M$ (‘C5’,’O9’)	$t \in R_M$ (‘C4’,’O10’)	$t \in R_M$ (‘C4’,’O8’)

Fig. 4.9 Tuple categorization in *R* for the join case3

In Fig. 4.9, (‘C_i‘, ‘O_j’) refers to the identifier for tuples shown in Fig. 4.7. For instance, when $t_1 \in S_{1A}$ (e.g., tuple with Cust_ID=’C1’ in S_1) and $t_2 \in S_{2A}$ (e.g., tuple with Order_No=’O1’ in S_2), then $t \in R_A$ (i.e., the tuple identified by (‘C1’,’O1’) in *R* is also accurate). Note that the tuple identified by (‘C2’,’O3’) is a mismember in *R* because of inaccurate value in a non-identifier attribute (i.e., S_2·Cust_ID). If the actual value for S_2·Cust_ID were recorded (say ‘C3’), then the tuple identified by (‘C2’,’O3’) in S_{temp} would have not been selected into *R* as a mismember but as an inaccurate. The tuple identified by (‘C3’,’O12’) belongs to R_C also because of the inaccurate value for its non-identifier attributes (i.e., S_2·Cust_ID). If the actual value for S_2·Cust_ID were recorded (say ‘C1’), then the tuple identified by (‘C3’,’O12’) in S_{temp} would have been selected into *R* as an accurate not a mismember. These results hold when the join condition applies to the entire identifier attributes of S_1 and a subset of the non-identifier attributes of S_2.

4.3 Quality Metrics for the Natural Join Operation

The natural join operation can be viewed to comprise of the following three stages:

i) Obtain the Cartesian product of S_1 and S_2 and store the result in S_{temp1}. Note that the combination of identifiers of S_1 and S_2 form the identifier for S_{temp1};

ii) Apply selection to S_{temp1} to select those tuples whose values agree in the common attributes between S_1 and S_2 and store the result in S_{temp2}; and

iii) For each common attribute in S_{temp2} (i.e., between S_1 and S_2), project out the corresponding attribute in S_2. The result is *R*.

The quality profiles for S_{temp1} and S_{temp2} are obtained as discussed for the θ-Join with the caveat that only the results of selection with arithmetic operator ’=’ must be applied to S_{temp1} since θ is always ‘=’ for the natural join. Relation *R* is obtained by applying the projection operation to project a subset of the attributes in S_{temp2}. Of importance here is the fact that there are no changes in the status of tuples after projecting out these attributes. This implies that *R* and S_{temp2} have the same quality profiles, i.e.,

$\alpha_R = \alpha_{temp2}$; $\beta_R = \beta_{temp2}$; $\mu_R = \mu_{temp2}$; and $\chi_R = \chi_{temp2}$.

5. A Numerical Example

We use our example for join case 2 to demonstrate how the quality profiles of R are obtained numerically. For this, we consider the quality profiles for the base relations shown in Figure 5.1.

	$\lvert\cdot\rvert$	α	β	μ	χ	q
S_1	5000	0.70	0.20	0.10	0.05	10
S_2	2000	0.80	0.10	0.10	0.12	15

Fig. 5.1 Quality profiles of the base relations

In addition, we suppose that the cardinality of the query output is given as $\lvert R \rvert = 3*10^6$. First, we obtain the quality profiles for S_{temp} (the Cartesian product of S_1 and S_2) using expressions (4.1) to (4.6). Next, we obtain the quality profiles for R using expression (4.7) to (4.10). The quality profiles for the query output are summarized in Figure 5.2.

	$\lvert\cdot\rvert$	α	β	μ	χ
S_{temp}	10^7	0.56	0.25	0.19	0.16
R	$3*10^6$	0.56	0.24	0.20	0.18

Fig. 5.2 Quality profiles for query result

We notice that the accuracy (inaccuracy) of R is less (greater) than the accuracy (inaccuracy) of either S_1 or S_2 (an implication of the Cartesian product operation). Mismembership and incompleteness of R are higher than those of S_1 and S_2. This is attributed to transformation of some of the inaccurate tuples in S_{temp} to mismembers and incompletes when they are selected into R.

In order to observe the effect of quality profiles of the base relations on those of the output relation, we perform a sensitivity analysis in respect to parameter α_2. For this, we fix α_1, β_1, μ_1, χ_1, μ_2, and χ_2 as given in Fig. 5.2. We change α_2 from 0.00 to 0.90 in steps of 0.01, and show the simulation result in Fig. 5.3. We notice that for the entire range of α_2, α_R is smaller than α_2 (and α_1) which is largely attributed to the effect of the Cartesian product operation. For low values of α_2, β_R is smaller than β_2 but as α_2 increases β_R becomes greater than β_2. Further, for the entire range of α_2, μ_R is greater than μ_2 (and μ_1) and χ_R is greater than χ_2 (and χ_1). This is because a proportion of the inaccurate tuples in S_{temp}contribute to mismember (incompletes) tuples in R (R_C).

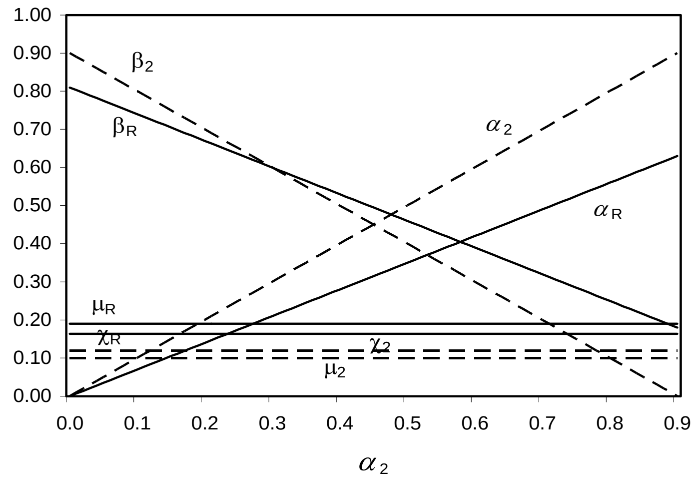

Fig. 5.3 Sensitivity analysis for α_2

6. Conclusions

In this research, we presented a methodology to assess the quality profiles for the output of the composite relational operation join. Specifically, we discussed the quality profiles for the output of join variants. These queries include more than one of the primitive operations selection, projection, and the Cartesian product. We show how these quality profiles can be obtained by applying the quality profiles of the Cartesian product followed by the applicable selection case. We also worked out a numerical example to demonstrate how our metrics work to assess the information quality. The work in this research can be further extended to investigate other types of join operation such as the outer join and its variants (left and right).

References

[1] Ballou D., Wang R., Pazer H., Tayi G., "Modeling Information Manufacturing Systems to Determine Information Product Quality", *Management Science* 44(4), Apr. 1998, pp. 462-484.

[2] Kon H., Madnick S., Seigel, M.D. "Good Answers from Bad data: A Data Management Strategy", *Proc. of WITS 1995.*

[3] Parssian A., Sarkar S., Jacob V. S., ""Assessing Data Quality For Information Products", *Working Paper, School of Management, University of Texas at Dallas*, May 2002.

[4] Redman T., "Data Quality for the Information Age", Artech House, 1996.

[5] Reddy M., Wang R., "Estimating data Accuracy in a Federated database Environment" *Proc. 6th Int'l Conf. on Information Systems and Data Management* 1995, pp. 115-134.

[6] Wand Y., Wang R., "Anchoring Data Quality Dimensions in Ontological Foundations", *Comm. ACM* 39(11), Nov. 1996, pp. 86-95.

[7] Wang R., Strong D., "Beyond Accuracy: What Data Quality Means to Data Consumers", *J. Management Information. System* (12(4), Spring 1996, pp. 5-34.