Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 577–584
ISBN 978-83-60810-27-9
ISSN 1896-7094

Trusted Data in IBM's MDM: Accuracy Dimension

(Work in progress)

Przemyslaw Pawluk
York University
Toronto ON, Canada
Center of Advanced Studies
IBM, Toronto
Email: pawluk@cse.yorku.ca

***Abstract*—A good data model designed for e-Commerce or e-Government has little value if it lacks accurate, up-to-date data [22]. In this paper data quality measures, its processing and maintenance in *IBM InfoSphere MDM Server* and *IBM InfoSphere Information Server* is described. We also introduce a notion of *trust*, which extends the concept of data quality and allows businesses to consider additional factors, that can influence the decision making process. In the solutions presented here, we would like to utilize existing tools provided by IBM in an innovative way and provide new data structures and algorithms for calculating scores for persistent and transient quality and trust factors.**

I. Introduction

MANY organizations have come to the realization that they do not have an accurate view of their business-critical information such as customers, vendors, accounts, or products. As new enterprise systems are added, silos are created resulting in overlap and inconsistency of information. This varied collection of systems can be the result of systems introduced through mergers and acquisitions, purchase of packaged applications for enterprise resource planning (ERP) or customer relationship management (CRM), different variant and versions of the same application used for different lines of business or home grown applications. Data in these systems typically differs both in structure and in content. Some data might be incorrect, some of it might just be old, and some other parts of it might show different aspects of the same entity (for example, a home vs. a work address for a customer).

Master Data Management (MDM) is an approach that decouples master information from the applications that created it and pulls it together to provide a single, unified view across business processes, transactional and analytical systems. Master data is not about all of the data of an organization. It is the data that deals with the core facts about the key entities of a business: customers, accounts, locations and products. Master data is high value data that is commonly shared across an enterprise – within or across the lines of business. MDM applications, such as IBM's InfoSphere Master Data Management Server, contain functionality to maintain master data by addressing key data issues such as governance, quality and consistency. They maintain and leverage relationships between master data entities and manage the complete lifecycle of the data and support multiple implementation approaches. MDM system itself is designed to support enterprise in the master data processing, integration and analysis.

As very important, quality of master data requires special attention. Different aspects or dimensions of quality need to be considered and maintained in all processes of the enterprise. *Trust scores*, introduced by this paper, can provide important information to the decision makers. Our approach to the quality of data is slightly different than described so far in the literature [4], [6], [9]. Our goal is to provide the user with the estimates of data quality and trust. Trust in this case is the aggregated value of multiple factors, and is intended to cover quality and non-quality aspects of master data. We are not making the attempts to build fixes nor enforce any quality policy. The information provided by us is intend to identify weaknesses of data quality or trustworthiness. The data quality enforcement should be then improved based on this information.

This paper focuses on the creation of measures, or trust factors, that serve to determine the trustworthiness of data being managed by MDM applications, specifically those being introduced in IBM's InfoSphere MDM Server. We would like to define here the model for data quality and methods of their processing in MDM. This new notion involves creating *trust scores* for trust factors that enhance the notion of data quality and the more broad quality-unrelated features such as lineage, security, and stewardship. All these have one goal – to support businesses in the decision making process, or data stewardship by providing information about different aspects of data. We would like to be more focus in this work on the quality aspects of data, especially the accuracy, which is one of the most commonly used quality dimension in the literature [2], [4], [12], [16], [20], [21], [26], [29], [31].

This paper is organized as follows. Section II presents the underpinning principles of Master Data Management (MDM), related concepts as well as the tools we used to prepare the trust scoring prototype. Section III provides a short overview of data quality and introduces the notion of trust. Section IV presents structures and methods used to acquire and store information about one of the most commonly used qualoity factor – accuracy. Section V describes our approach to accuracy estimation in SQL query processing.

II. MDM AND INFORMATION SERVER

Master data management is a relatively fast growing software market. Many customer acknowledge they have data quality and data governance problems and look for solutions to these problems. Crucial parts of such MDM solutions are data quality and data trust mechanisms [8], [10], [23]. In this section we present the MDM environment and the comprehensive approach to data trust and data quality that utilizes tools provided by IBM.

A. Definitions

Master Data Management (MDM) provides the technology and processes to manage Master Data in an organization. Master Data is the data an organization stores about key elements or business entities that define its operation. An MDM solution enables an enterprise to govern, create, maintain, use, and analyze consistent, complete, contextual, and accurate master data information for all stakeholders. Master data is typically high value information that an organization uses repeatedly across many business processes. For these to operate efficiently, this master data must be accurate and consistent to ensure correct business decisions. Unfortunately in many organizations, master data is fragmented across many applications, with many inconsistent copies and no plan to improve the situation.

Master Data Management (MDM) products differs from traditional approaches to the masted data, that include the use of existing enterprise applications, data warehouses and even middleware. It *is not* domain-centric approach such as CRM application for the customer domain or ERP application for the product domain. Some MDM products decouple data linked to source systems so they can dynamically create a virtual view of the domain, while others include the additional ability to physically store master data and persist and propagate this information. Some products are not designed for a specific usage style, while others provide a single usage of this master data. Even more advanced products provide all of the usage types required in today's complex business-collaborative, operational and analytic-as out-of-the-box functionality. These products also provide intelligent data management by recognizing changes in the information and triggering additional processes as necessary. Finally, MDM products vary in their domain coverage, ranging from specializing in a single domain such as customer or product to spanning multiple and integrated domains. Those that span multiple domains help to harness not only the value of the domain, but also the value between domains, also known as relationships. Relationships may include customers to their locations, to their accounts or to products they have purchased. This combination of multiple domains, multiple usage styles and the full set of capabilities between creating a virtual view and performance in a transactional environment is known as multiform master data management.

Achieving a high level of data quality is key prerequisite for many of the MDM objectives. Without high quality data the best analytics and business intelligence applications are still going to deliver unreliable input to important business decisions. Another key aspect of the management of the master data is achieving a high level of trustworthiness in the data. It is a key factor for customers to have reliable information about the data. Information about the quality, the origin, the timeliness and many other factors influence the business decisions based on the provided data.

The introduction of *data governance* in the organization is a vital prerequisite to come to more trusted information. Moving to master data management can be the cornerstone of a data governance program. It is important however to note that at the same time, moving to MDM cannot be successful without data governance.

Data governance is defined as "the orchestration of people, process and technology to enable an organization to leverage information as an enterprise asset" [15]. It manages, safeguards, improves and protects organizational information. The effectiveness of data governance can influence the quality, availability and integrity of data by enabling cross-organizational collaboration and structured policy-making.

B. MDM Tools

We will present here some tools provided by IBM that are very useful in terms of quality assessment and management. All of them are parts of the IBM InfoSphere platform.

IBM InfoSphere MDM Server is an application that was built on open standards and the Java Enterprise Edition(JEE) platform. It is a real-time transactional application with a service-oriented architecture that has been built to be scalable from both volume and performance perspectives. Shipping with a persistent relational store, it provides a set of predefined entities supporting the storage of master data applicable to each of the product's predefined domains. It also includes the MDM Workbench – an integrated set of Eclipse plug-ins to IBM Rational Software Architect/Developer that support the creation of new MDM entities and accompanying services, and a variety of extensions to MDM entities. *IBM InfoSphere Information Analyzer* that profiles and analyzes data so that the system can deliver trusted information to users. The Information Analyzer (IA) is used to scan or sample data stored in diverse sources to assess its quality. MDM also uses some complementary tools: *IBM InfoSphere QualityStage*, which allows us to define rules to standardize and match free-form data elements which is essential for effective probabilistic matching of potentially duplicate records, and *IBM WebSphere AuditStage*, which enables us to apply professional quality control methods to manage the accuracy, consistency, completeness, and integrity of information stored in databases. We also use statistics provided by *IBM InfoSphere DataStage* to compute chosen quality and trust factors.

This set of tools provides a comprehensive approach to data quality and data trust management. This approach not only resolves some problems during the data acquisition but also allows us to control the level of data trust and to give up-to-date information about the trustworthiness to a user. This comprehensive approach is novel. Moreover our solution does

not require any specialized hardware or operating system and is able to cooperate with any commercial data base systems.

1) IBM InfoSphere MDM Server: The InfoSphere Master Data Management Server has a new feature allowing users to define and add quality and trust factors to the data of their enterprise. This new data structure enables the user to store metadata required to compute scorings for trust and quality of data. Provided wizards allows user to modify the data model in a simple way.

2) IBM Information Server: IBM Information Server addresses the requirements of cooperative effort of experts and data analysts with an integrated software platform that provides the full spectrum of tools and technologies required to address data quality issues [1]. It supplies users and experts with the tooling that allows the detailed analysis of data through profiling (*IBM InfoSphere Information Analyzer* and *IBM InfoSphere AuditStage*), cleansing (*QualityStage*) and data movement and transformation (*DataStage*). In this paper we concentrate on data profiling and analysis handled mostly by Information Analyzer (IA), AuditStage (AS), and partially on QualityStage (QS).

IA, as an important tool of *data quality assessment* (DQA) process, aids the exposing technical and business issues. The technical issues detection is a simpler part of the process based on technical standards and covers important problems including different or inconsistent standards in structure, format, or values, missing data and default values, spelling errors, data in wrong fields, and buried information in free-form fields. Business quality issues are more subjective and are associated with business processes such as generating accurate reports. They require the involvement of experts.IA helps the expert in systematic analysis and reporting of results, thereby allowing him to focus on the real problem of data quality issues. This is done through tasks like column analysis, key analysis(Primary and Foreign Key) and cross-table analysis.

a) QualityStage: IBM InfoSphere QualityStage (QS) complements IA by investigating free-form text fields such as names, addresses, and descriptions. QS allows users to define rules for standardizing free-form text domains which is essential for effective probabilistic matching of potentially duplicate master data records. It provides user with functions such as free-form text investigation, standardization, address verification and record linkage and matching as well as survivorship that allows best data across different sources to be merged.

b) AuditStage: IBM WebSphere AuditStage (AS) enables user to apply professional quality control methods to manage different subjective quality factors of information stored in databases such as accuracy, consistency or completeness. By employing technology that integrates Total Quality Management (TQM) principles with data modeling and relational database concepts, AS diagnoses data quality problems and facilitates data quality improvement effort. It allows performing assessment of the completeness, validity of critical data elements and business rule compliance. AuditStage is very useful tool for assessment of the *consistency* factor allowing cross-table rules validation.

III. The Notion of Trust

Trust is an extension of data quality. We are looking for additional factors because data quality is not the only factor influencing the trustworthiness of data and these two concepts are not necessarily correlated. Low-quality data may be trusted in some situations and high-quality data may have low trustworthiness in other. The value of trust strongly depends on the user requirements and usage context. In this section we discus a data quality and we describe the notion of trust.

A. Data Quality

The data quality concept has been widely discussed in literature [2]–[4], [6], [9], [11], [19]–[21], [25], [27], [28], [32], [33] usually in the context of a single data source. Some work tough has been also done in the context of integrated data [5], [7], [12], [13], [18], [24] emphasizing the importance of data quality assurance in this area. Batini and Scannapieco [4] have given three examples of organizational processes where DQ aspects are particularly important.

- Customer matching – it is a common issue in organizations where more than one system with overlapping databases exists. A typical result is an inconsistent and duplicate information.
- Corporate house-holding – is a problem of identifying members of household (or related group). This context-dependent issue is widely described in [30].
- Organization fusion – is the issue of integration legacy software in case of organizations or units merge.

The definition of the quality that we are using in this work originates from the one provided by Naumann [18] as an attempt to provide an operational definition of DQ as an aggregated value of multiple IQ-criteria (Information Quality Criteria). IQ-criteria are there classified into four sets:

- Content-related – intrinsic criteria, concerned with the retrieved data,
- Technical – criteria measuring aspects determined by software and hardware of the source, the network and the user,
- Intellectual – subjective aspects of data that shall be projected to the data in the source,
- Instantiation-related – criteria related to the presentation of the data.

We follow the Naumann's approach by defining data quality as a aggregated value of multiple DQ-factors. Later we will extend this definition introducing the trust notion.

B. Trust Definition

Following Naumann's definition of data quality, we define trust (data trust, DT) through factors that influences the trustworthiness of data.

Definition 3.1 (Data Trust): Trust is the aggregated value of multiple Data Trust factors.

This definition provides flexibility when defining trust for a specific industry and user requirements. The trust factor (DT-factor) may be a DQ-factor, or non-quality (NQ) factor.

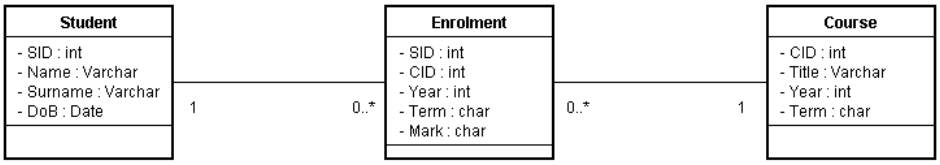

Fig. 1. Sample database schema representing enrollment of students into courses

Here we concentrate on data quality factors, and especially on accuracy dimension. However other factors like data lineage, security or trust of data source can be considered.

C. Accuracy dimension

Accuracy is included by most data quality studies as a key factor [2], [4], [12], [16], [20], [21], [26], [29], [31]. Although the term has an intuitive appeal, there is no commonly accepted definition of what it means exactly [29]. Ballou and Pazer [2] describe accuracy as "the recorded value being in conformity with the actual value." Kriebel [16] characterizes accuracy as "the correctness of the output information." Thus, it appears the term is viewed as equivalent to correctness.

In [4] accuracy is defined as "the closeness between a value v and a value v', considered as the correct representation of the real-life phenomenon that v aims to represent." The simple example can be the name of the city $Toronto$, the value $v = Tronto$ is incorrect (inaccurate) and $v' = Toronto$ is correct (accurate).

Parssian et al. [20], [21] formalize the notion of accuracy and propose quality metrics to assess the quality of basic queries. However definition proposed by Parssian et al. describes accuracy on the higher level of granularity. They consider the tuple as a whole, we in a contrast considering each attribute separately.

The accuracy may be calculated in three possible ways:

1) Inspection by expert – the expert reviews records and marks inaccurate ones;
2) Automatic or Semi-Automatic inspection – the system refine set of candidate records based on some predefined set of rules. Expert may or may not review and correct the result;
3) As a distance function utilizing dictionary sets – we may calculate the minimal number of operation required to transform the value v into the closest dictionary entry v'. Based on that calculation experts may identify low quality entries and apply changes to improve quality.

In all cases we consider only measurement. Experts and system is not allowed to perform any changes in data content. Moreover the (semi-)automatic inspection may be done only if some patterns apply to the field or dictionary may be defined. Examples of fields for which we can define some accuracy rules are SIN, postal code or phone numbers where some common rules apply. The dictionary, sometimes called code table, may be defined for fields like city, country etc. However, there are fields where we are unable to define neither rules nor dictionary. I.e. notes defined as a text field may contain any string. Moreover it is not possible to confront in the automatic way the stored data with real world to confirm the accuracy.

The accuracy and inaccuracy of data may be perceived in many different ways depending on the domain. At this point we will consider two approaches *error bar* and *boolean*. The first one, borrowed from engineering, gives some flexibility but is applicable only to ratio- or interval-scaled types [14]. The second one considers the value strictly in "black and white" – it may be accurate or inaccurate, one or zero, and there is nothing in between.

1) Error bar: In many areas of science, especially in physics and engineering, the metrology defines a way of calculation of measurements' error and its propagation in calculations. We can see data stored in database as such measurements' results taken with some error and can define *inaccuracy* of data as the relative error. Value of this error is normalized. Such definition applies to numerical fields, ratio- and interval-scaled, where the distance between two values is meaningful, and expresses the relative distance between stored value and real world value. For text fields we can define distance using number of atomic changes required to transform our value into real world value divided by length of real world value.

Accuracy defined in this way however is applicable only to limited types of attributes (preferably numerical) and in specific domains. In general this kind of interpretation is inapplicable in database environment, where we are not provided with any information about inaccuracy of stored data.

2) Boolean accuracy: Boolean approach originates from the idea or rather assumption that each stored record has some source in the real world, and that this source can be "linked" in an unambiguous way to the stored tuple. In such case we consider the value accurate if the value of stored instance is equal to the value of its real world source. This approach bounds the instance and source, however does not provide any flexibility. The stored instance is either accurate or inaccurate, and there is nothing in between of those extremes, even though this approach can be applied widely.

3) Hybrids: The third approach merges two ideas. We allow our stored instance to vary slightly from the real world source but the result remains boolean. The stored value v is considered accurate in this case if $|v - v'| \leq \epsilon$, where v' is a real world value and ϵ is some predefined, acceptable variation. This variation may also be expressed as a relative value (i.e. percentage). Then value is accurate if $\frac{|v-v'|}{v'} \leq \epsilon$.

In the following consideration we restrict the calculation methods to boolean interpretation of the accuracy; however

TABLE I
SAMPLE DATA IN TABLES STUDENT, COURSE AND ENROLLMENT FROM THE DATABASE FROM FIGURE 1. GRAY FIELDS CONTAIN INACCURATE VALUES

Student

CID	Name	Surname	DoB
111111111	AAAAAAAA	AAAAAAAAAA	1971-01-01
222222222	BBBBBBBB	BBBBBBBBBB	1981-01-01
333333333	CCCCCCCC	CCCCCCCCCC	1973-05-02
444444444	DDDDDDDD	DDDDDDDDDD	1971-09-10

Course

CID	Title	Year	Term
CSE2222	Software Tools	2009-10	F
CSE2222	Software Tools	2009-10	W
CSE2222	Software Tools	2009-10	S
CSE1111	Data Bases	2009-10	F

Enrollment

SID	CID	Year	Term	Mark
111111111	CSE2222	2009-10	F	A
222222222	CSE1111	2009-10	F	B
333333333	CSE2222	2009-10	F	A
222222222	CSE2222	2009-10	W	C

hybrid interpretation may be used as well. The only requirement is that method should be defined before, so we can use it to determine the accuracy of data.

D. Accuracy – Definitions

The base concept of the accuracy here is the confrontation of stored values with their real world sources. We assume that each tuple can be unambiguously linked with the real world element through the accurate key. If we can identify this real world element, then we can compare stored values and real world values to determine the accuracy of stored values. This real world element is called *source*. The linkage with real world entity is a key property in accuracy semantic and is necessary to provide the interpretation of derived accuracy values. We are using the distinction between keys and measures used commonly in warehouses. We would like to provide the definitions for both those types. The key attribute can be seen as a link connecting stored data with source entities. The ability of unambiguous identification of the source by the key is the base for accuracy calculation.

Example 1 (Accuracy interpretation): Let's consider the sample schema presented on Figure 1. This schema consists of three relations:

- $Student(\underline{SID}, Name, Surname, DoB)$
- $Course(\underline{CID}, \underline{Year}, \underline{Term}, Title)$
- $Enrolment(\underline{SID}, \underline{CID}, \underline{Year}, \underline{Term}, Grade)$

There are defined also two foreign key constraints on the table Enrollment:

- FK(SID) – where SID is a key in Student relation
- FK(CID, Year, Term) – where CID, Year and Term are compound key in Course relation

The accuracy of any field in Student relation can be determined only if we can determine the real world source. This determined value can be also interpret only in presence of the key. For example, let's consider student with SID=111111111. The accuracy of the Surname can be determined only by comparison of stored value and source value. If the source value is different than the stored one we say that stored value is inaccurate.

Now considering student record with SID=333333333, which is marked as inaccurate, we are not able to link this entry with source. In such situation we cannot compare stored values with source, because source is unknown.

Base on the above example we see that we have to distinguish two types of accuracy:

- accuracy of key – determines the accuracy of the entire row,
- accuracy of measure – is determined by two factors: the accuracy of the key and the source (by comparison of the stored and source values)

In this example we also see that inaccurate key breaks the linkage between stored entity and source entity making impossible the assessment of the accuracy of measures.

Definition 3.2 (Accuracy of key): The key attribute is accurate if and only if it unambiguously identifies the source object. If the key is a composite the accuracy is calculated for the entire key, and is inherited by each key element.

Example 2 (Accuracy of key): Let's consider relations `Student` and `Curse` presented by Table I. Relation `Student` has a key attribute `SID`. It can be the same number that appears on the students id and it allows us to identify a student in unambiguous way. If we cannot identify the student (i.e. there does not exist student with chosen SID) all data related to this key will be inaccurate. Relation `Course` has a compound key that consists `CID`, `Year` and `Term`. This triple allows us to identify courses. If one of those elements is inaccurate (i.e. term is equal `'T'` which is incorrect value), entire key is inaccurate and all data related to this key is considered as inaccurate. Moreover, if `SID` or one or more key elements from `Course` are inaccurate, then all related entries in `Enrollment` will be inaccurate.

The accuracy of measure is dependent on the accuracy of the key. The accuracy of the measure is based on the idea that the data to be accurate should match source data.

Definition 3.3 (Accuracy of measure): We consider the measure accurate if and only if the key of the tuple ($x.X_0$) is accurate and measure's value ($x.X_i$) is equal to the real world value ($x'.X_i$) identified by the key of the tuple that both key and measure belong to.

$$Acc(x.X_i) = \begin{cases} 1 & \text{if} \quad x.X_i = x'.X_i \wedge Acc(x.X_0) = 1 \\ 0 & \text{if} \quad x.X_i \neq x'.X_i \vee Acc(x.X_0) = 0 \end{cases} \tag{1}$$

IV. Acquisition and Processing of Accuracy

Trust and quality processing described below is one of the most novel aspects of our work. An important advantage of our approach is the use of existing set of tools, slightly modified or extended to serve in the new context. We extended these tools by creating data structures to store and process meta-data describing data quality and trust. We have designed mechanisms for assessing accuracy of data. Our approach is view-based estimation of accuracy of a SQL answer.

In this section we will explain how the information about the accuracy of data is gathered and processed. We will use the view-based model to estimate the accuracy of data in each view's attribute, then the estimated accuracy will be inherited by view's attribute elements.

A. Quality Data Structures

MDM provides a mechanism that enables an extension of the existing data with trust/data quality factors. These extensions may be defined as *persistent object* and stored in the database or be as *transient objects* calculated at run time. We would like to use this mechanism and employ it to tag stored data with the accuracy score.

B. Views

Thinking about the accuracy of data we can easily notice that checking each value and confronting it with the source is not applicable approach, especially in the context of huge governmental or commercial databases. From the other hand big picture defined by some general statistical analysis is often not enough. Our idea inspired by Motro and Rakov's work [17] is employing views over base tables to represent external knowledge about the quality of data.

Definition 4.1: View is a intentional or extensional set of tuples chosen from base table.
In other words the view can be defined by specifying a query or by specifying an extension – set of tuples – by pointing directly members of this set.

Views can overlap. It means that one tuple from the base table can be a member of one or more view. We assume also that each tuple is a member of at least one view. This assumption can always hold, because the entire base table can be seen as one, very general view.

We preserve the relational semantic of view, which can be seen as a set of attributes (in exactly the same way as a table). For each attribute then, the accuracy metric can be calculated. This metric expresses the statistical likelihood of choosing accurate value. The value of this metric is inherited by all elements of the attribute.

Value of the accuracy metric is calculated base on accuracy measure. When operating on sufficiently small set, we can test each value to calculate metric, otherwise sampling and statistical methods can be used to determine the likelihood of choosing accurate value.

The view overlapping rises some problems. Without detailed knowledge of error distribution we are not able to assess the accuracy of the intersection of two views. As an intersection we understand a set that is a subset of both views. In such case we assume that smaller view provide us with more detailed information hence we will use its metrics to tag tuples from the intersection.

Example 3 (View-based accuracy): Let's consider the relation Course (Figures 1 and I). Let's also assume that we have information that the accuracy of entries inputed for the summer term 2009-10 have lower accuracy than average. Base on that knowledge we define view V containing all courses having Term='S' and Year='2009-10'. For each attribute we calculate two metrics: one general, which covers entire attribute, the second one over view V. Each value of the attribute is tagged with appropriate value – if it is covered by view V, it is tagged with metric calculated for the view's attribute, if not, the general metric is used.

C. View elements tagging

In previous subsection we have defined accuracy metrics, which are calculated for each attribute of the view. Here we will explain how view's elements are tagged with the accuracy scores (value of accuracy metric).

Let's remind the assumptions we have made: (1) Views may overlap; (2) Inaccurate elements are distributed over attributes in uniform way. Value of the accuracy calculated for the attribute of the view is inherited by each element of the attribute. Those tags can be aggregated at the end of the processing. The average calculated over all tags will be an equivalent to the weighted average over views. We will use values assigned to elements rather than view in the estimation of the accuracy of query answer.

V. Query processing with accuracy

The trust alone is just yet another piece of data given to the user. The really important question is *What can be done with this information?* Lets consider now some use cases showing usage of the trust in the system.

We have shown that the trust score can be incorporated in our meta-data and linked with each field in the database if desired. This information can be then returned to the user. Even though this information is very detailed, it is not practically useful in all cases. Without algorithms to propagate trust in the query processing, we can only annotate a tuple and return it to the user. However we can build some statistics over this information that can be used later.

One of the problems that are currently unsolved is propagation of trust scores in the query processing. We are currently working on methods allowing us to estimate the trust of the result of the SQL operator based on the estimated trust of entry set. We are using estimates in this context because it is significantly less expensive than reaching out each time for the data.

Analyzing the quality of the query response, the key operation to be considered is selection. All other operations, except projection, rely somehow upon selection. For example *join* operation can be expressed as Cartesian product followed by

TABLE II
DIFFERENT CASES OF THE ACCURACY DEPENDING ON THE AGGREGATION AND TYPE OF SELECTION

	With aggregation		**Without aggregation**
	Count	Other	
Non-trivial	$Acc(X_i) \cdot 1$	$Acc(\sigma) \cdot Avg(Acc(X_i))$	$Acc(\sigma) \cdot Acc(X_i)$
Trivial	$1 \cdot 1$	$1 \cdot Avg(Acc(X_i))$	$1 \cdot Acc(X_i)$

selection and *group-by* operation can be seen as a bunch of selections followed by aggregations.

The accuracy of measure can be determined and interpreted only in the presence of the accurate key. Because of this strong relation between key and accuracy of measures, the latter can be seen as functionally dependent on the key.

Definition 5.1: The accuracy context is a set of key attributes allowing for an unambiguous assignment of accuracy for a field of specific entity or aggregate over a set of entities.

Example 4 (Context preservation): Let's consider the relation Student (Figure I.) and two queries:

- `SELECT * FROM Student`
- `SELECT Name, Surname, DoB FROM Student`

Both queries return all records from the relation Student. The first query however returns entire relation (all attributes) when the second query eliminates the key of the relation (SID). As a result of the elimination of SID from the answer we are unable to connect the arbitrary element from the output with its source entity, hence we cannot calculate the accuracy.

A. Considered query types

In this work, the following types of queries will be considered:

- equi-selects – we consider the equi-select with the key of selection being an attribute which has nominal or ordinal type, preferably a relation key;
- range selects – can be done over attributes which have ratio- or interval-scaled type (the distance between values has a meaning) or over ordinal attributes (the order has a meaning), but not over nominal attributes;
- select with aggregation – both equi- and range select can consist the aggregation. We have found two subsets of this type of queries:
  - count – the value of the attribute (if not NULL) does not influence the result;
  - other aggregates – values have an influence on the result, the scale of the impact depends on the aggregate function.

When considering range selects following problem arises: some ranges may be defined in a way that the answer is entire relation. We consider this class of queries because even though is seems that some comparison is required to determine if tuple should be returned, in fact, no comparison is necessary.

Example 5 (Trivial range selection): Let's consider the relation Student and following range query:

```
SELECT * FROM Student
WHERE
DoB BETWEEN 1900-01-01 AND 2100-12-31
```

The range of DoB is from 1971-01-01 to 1981-01-01. We clearly see that the WHERE clause of the query covers entire relation and because of that the query is semantically equivalent to the query `SELECT * FROM Student`. In such case, comparisons are not necessary to derive the answer.

B. Accuracy components

We have identified two main components of the accuracy of the selection's result. The first one is the accuracy of the attribute factor, denoted as $Acc(X)$ where X is an attribute. Including this component seems to be natural and obvious. The second component is the accuracy of selection, denoted as $Acc(\sigma)$, which expresses the likelihood that an arbitrary tuple has been accurately selected. It is the likelihood that the arbitrary element from the selection key is accurate. This component has to be covered and propagated over all derived attributes.

We have to explore how those components interfere in different types of queries. It can be easily noticed that selection component will not appear in trivial selections since we cannot make any mistake. On the contrary in the query employing counting operation the accuracy of attribute being counted does not meter because we are interested only in the number of elements, not the exact values. Table II gathers all cases that we have identified. It also presents our preliminary proposition of calculation of the accuracy of the selection's result. It is based on the assumption that attributes are independent in terms of errors' distribution. In such case we can see the accuracy of derived element as a result of two independent events: accurate selection and accurate value of the considered attribute.

VI. CONCLUSIONS

Measuring data quality and data trust is one of the key aspects of supporting businesses in decision making process or data stewardship. Master Data Management in other hand supports sharing data within and across lines of business. In such case trustworthiness of the shared data is extremely important. Our investigation has resulted in consistent method of gathering and processing quality and trust factors.

In this work we have presented the *IBM InfoSphere MDM Server* and elements of *IBM Information Server* such as DataStage, QualityStage, AuditStage and Information Analyzer, and their ability to handle data quality and data trust. We have also presented the new notion of data trust. The process of gathering and computing data quality and trust factors has been described and explained using example.

This work covered only the introduction to the accuracy processing. We consider select as a most basic operation, which is necessary to proceed with other operations and though should be explored soon. We have proposed the data model for the accuracy storing and processing. In the future we are going to cover in our consideration all SQL operators including accuracy neutral operators (Cartesian product, projection) and other accuracy sensitive operators (join, group-by and set operations). We suspect that projection and Cartesian product are accuracy-neutral because those operations does not relay upon accuracy of input. They do not "touch" data values.

ACKNOWLEDGMENTS

Author would like to thank Jarek Gryz and Parke Godfrey from York University for support and constructive criticism, Stephanie Hazlenwood and Paul van Run from IBM Toronto Lab for support and valuable discussions.

This work has been supported by IBM Center for Advanced Studies and NSERC.

REFERENCES

[1] ALUR, N., JOSEPH, R., MEHTA, H., NIELSEN, J. T., AND VASCONCELOS, D. *IBM WebSphere Information Analyzer and Data Quality Assessment*. Redbooks. International Business Machines Corporation, 2007.

[2] BALLOU, D., AND PAZER, H. Modeling data and process quality in multi-input, multi-output information systems. *Management Science 31*, 2 (1985), 150–162.

[3] BALLOU, D., WANG, R., PAZER, H., AND TAYI, G. K. Modeling information manufacturing systems to determine information product quality. *Manage. Sci. 44*, 4 (1998), 462–484.

[4] BATINI, C., AND SCANNAPIECO, M. *Data Quality: Concepts, Methodologies and Techniques*. Data-Centric Systems and Applications. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] BOUZEGHOUB, M., AND KEDAD, Z. *Quality in Data Warehousing*. Kluwer Academic Publisher, 2002.

[6] CROSBY, P. B. *Quality is free : the art of making quality certain / Philip B. Crosby*. McGraw-Hill, New York :, 1979.

[7] CUI, Y., WIDOM, J., AND WIENER, J. L. Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst. 25*, 2 (2000), 179–227.

[8] DREIBELBIS, A., HECHLER, E., MATHEWS, B., OBERHOFER, M., AND SAUTER, G. Master data management architecture patterns. http://www.ibm.com/developerworks/data/ library/techarticle/dm-0703sauter/index.html, 2007.

[9] ENGLISH, L. Information quality improvement: Principles, methods, and management. Seminar, 1996. 5th Ed., Brentwood, TN: INFORMATION IMPACT International, Inc.

[10] FAN, W. Dependencies revisited for improving data quality. In *PODS '08: Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2008), ACM, pp. 159–170.

[11] FOLEY, O., AND HELFERT, M. The development of an objective metric for the accessibility dimension of data quality. In *Proceedings of International Conference on Innovations in Information Technology* (Dublin, 2007), IEEE, pp. 11–15.

[12] GERTZ, M., AND SCHMITT, I. Data Integration Techniques based on Data Quality Aspects. In *Proceedings 3. Workshop "Föderierte Datenbanken", Magdeburg, 10./11. Dezember 1998*, I. Schmitt, C. Türker, E. Hildebrandt, and M. Höding, Eds. Shaker Verlag, Aachen, 1998, pp. 1–19.

[13] GUPTA, A., AND WIDOM, J. Local verification of global integrity constraints in distributed databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993* (1993), P. Buneman and S. Jajodia, Eds., ACM Press, pp. 49–58.

[14] HAN, J., AND KAMBER, M. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2001.

[15] IBM. Ibm master data management: Effective data governance. ftp://ftp.software.ibm.com/software/uk/itsolutions/information-management/information-transformation/ master-data-management/master-data-management-governance.pdf, 2007.

[16] KRIEBEL, C. H., AND MOORE, J. H. Economics and management information systems. *SIGMIS Database 14*, 1 (1982), 30–40.

[17] MOTRO, A., AND RAKOV, I. Not all answers are equally good: estimating the quality of database answers. 1–21.

[18] NAUMANN, F. *Quality-driven query answering for integrated information systems*. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[19] OLSON, J. *Data Quality: The Accuracy Dimension*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[20] PARSSIAN, A., SARKAR, S., AND JACOB, V. S. Assessing information quality for the composite relational operation join. In *IQ* (2002), pp. 225–237.

[21] PARSSIAN, A., SARKAR, S., AND JACOB, V. S. Assessing data quality for information products: Impact of selection, projection, and cartesian product. *Manage. Sci. 50*, 7 (2004), 967–982.

[22] RADCLIFFE, J. Magic quadrant for master data management of customer data. Tech. Rep. G00167733, Gartner, Inc., 2009. http://mediaproducts.gartner.com/reprints/ oracle/article78/article78.html.

[23] RADCLIFFE, J., AND WHITE, A. Key issues for master data management. Gartner Master Data Management Summit, Chicago, IL, 2008.

[24] REDDY, M. P., AND WANG, R. Y. Estimating data accuracy in a federated database environment. In *CISMOD* (1995), pp. 115–134.

[25] REDMAN, T. C. *Data quality : the field guide*. Digital Pr. [u.a.], Boston, 2001.

[26] SHIN, B. An exploratory investigation of system success factors in data warehousing. *J. AIS 4* (2003).

[27] TAYI, G. K., AND BALLOU, D. P. Examining data quality. *Commun. ACM 41*, 2 (1998), 54–57.

[28] TUPEK, A. R. Definition of data quality, 2006.

[29] WAND, Y., AND WANG, R. Y. Anchoring data quality dimensions in ontological foundations. *Commun. ACM 39*, 11 (1996), 86–95.

[30] WANG, R. Y., CHETTAYAR, K., DRAVIS, F., FUNK, J., KATZ-HAAS, R., LEE, C., LEE, Y., XIAN, X., AND BHANSALI, S. Exemplifying business oppurtunities for improving data quality from corporate household research. In *Advances in Management Information Systems - Information Quality (AMIS-IQ) Monograph* (April 2005).

[31] WANG, R. Y., PIERCE, E. M., AND MADNICK, S. E. *Information quality*, vol. 1 of *Advances in management information systems: Information Quality*. M.E. Sharpe, 2005.

[32] WANG, R. Y., AND STRONG, D. M. Beyond accuracy: what data quality means to data consumers. *J. Manage. Inf. Syst. 12*, 4 (1996), 5–33.

[33] YANG, L. P., LEE, Y. W., AND WANG, R. Y. Data quality assessment. *Communications of the ACM 45* (2002), 211–218.