Warehouse Builder 11*g*

Matching and Merging data – Black Art or Exact Science

April 2008

NOTE:

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Matching and Merging data – Black Art or Exact Science

INTRODUCTION

Enterprises have always relied on data to be successful. Customers, products, suppliers, and sales transactions all need to be described and tracked in one way or another. Even before computers became commercially available, the data in the form of paper records has been vital to both commercial and non-commercial organizations. With the advent of computing technology, the sophistication of data usage by businesses and governments grew exponentially. The technology industry serving these needs has generated many buzz words that come and go: decision support systems, data warehousing, customer relationship management, business intelligence, etc., but the fact remains the same—organizations need to make the best use of the data they have to increase their efficiency today and improve their planning for tomorrow.

If the data is so fundamental to the business, it is not surprising that a lot of effort goes into acquiring and handling data and making it available to those who need it. In the process, the data is moved around, manipulated, and consolidated. The quality of the data is rarely a high priority as the pressure is on to deliver the project and “get the data out to the users.” The justifications for not making data quality a priority are often just thoughts such as “our data is good enough” and “we can always clean it later.”

One of the biggest stumbling blocks in today’s environment, apart from just really bad data, is the matching and merging of disjoint data elements. Being the data acquired through mergers and acquisitions or by consolidating systems, many organizations face the problem of making sense of duplicate (or not) data.

In this paper we will discuss data quality in general, but then focus on matching and merging data with Oracle Warehouse Builder 11g.

WHAT IS DATA QUALITY?

Data quality is an all-encompassing term describing both the state of data that is complete, accurate, and relevant, as well as the set of processes to achieve such a state. The goal is to have data free of duplicates, misspellings, omissions, and unnecessary variations, and to have the data conform to the defined structure.

Simply put, the data quality addresses the problem cynically but precisely summed up as “garbage in-garbage out.”

A significant part of data quality deals with customer data—names and addresses, due to both their key roles in business processes and their highly dynamic nature. Names and addresses are ubiquitous—they tend to exist in almost every source and are often the only identifying data. Most matching applications rely heavily on names and addresses, because a common unique identifier is typically not available across systems. Consequently, whatever data is available must be used to determine if different individuals, businesses, or other types of records are actually the same. But names and addresses usually contain dirty data, since they often include nicknames, abbreviations, misspellings, bad fielding, etc. Furthermore, the name and address data consistently deteriorates over time as people move and/or change last names. Sophisticated name and address processing can solve these problems and have a significant positive effect on matching.

The focus of data quality on names and addresses sometimes causes a misconception that data quality is just about ensuring that names and addresses are correct and thus postal mailings will be deliverable to customers. Therefore, the thinking goes, if your business does not send bills or orders to customers by mail, the data quality is not that important. This is wrong for two reasons:

1. The correct and standardized name and address data is not the end goal, or not the only goal. The end goal is to identify and match customers reliably based on name and address data.
2. The data quality is certainly not limited to names and addresses. Any data, such as product data, will benefit from being standardized and complete.

DATA QUALITY IS CRITICAL

There are many challenges in building a data warehouse or performing any data integration project. Consequently there are many areas of importance for the tools that assist with building a data warehouse. Is it easy to build and maintain the data warehouse (productivity and maintainability)? Can all required sources of data be brought into the data warehouse (source integration)? Can the data warehouse continue to add data within allowed time periods and can it accommodate the growth (performance and scalability)?

However, to the end users of the data warehouse making the business decisions based on the data in the data warehouse, the most important question is “can I trust this data?” It doesn’t matter if the data warehouse was built in record time, it practically maintains itself, every imaginable source system is in, and the loads are blazingly fast and are only getting faster. If the business users find the data in the data warehouse not to be trustworthy, they will not use it to make their business decisions. All the impressive achievements in those other areas will then be meaningless.

Having complete, accurate, and relevant data free of all defects sounds great on paper but is never entirely achieved in reality. Instead, each organization defines, formally or informally, the acceptable level of data quality; i.e., a certain threshold that can be measured. Just as impressive are the figures that show the losses to businesses due to poor data quality and the estimates on how much is spent on trying to address the data quality issues. Therefore, it is not surprising that the true goal for businesses is not the absolute data quality in its academic sense but the acceptable level of data quality achieved efficiently in terms of time, effort, and money.

INTEGRATING DATA QUALITY INTO ETL

For anyone who follows the data integration industry through the press, analyst reports, or vendor events, there should be nothing new so far. The definition of data quality and the acknowledgement of its importance are universal in the industry.

Compared to other solution, the key difference of data quality in Warehouse Builder is the level of integration into the ETL process, making data quality transformations an inseparable function of data warehouse development.

Consider an analogy with a spell checker in a word processor. It would hardly be acceptable to anyone to close the document, start a spell check application, process and correct the document, then re-open it in the word processor—even if the spell checker has a great user interface of its own, is fast, and is otherwise pleasant to work with. Why not just invoke the spell checker from within a word processor? Better yet, why not correct the mistakes as the document is created? Of course there is no such problem in modern word processors. Spell checkers have long been integrated and work seamlessly. Another interesting parallel to note is that it's fairly unimportant to the user of a word processor that the actual spell checking technology is provided by a third-party vendor.

So within Warehouse Builder, data quality is built-in, using the same interface for design and management of data quality processes as for the ETL processes, just like your word processor and its spell checker.

Further, Warehouse Builder offers product-wide services that data quality transformations benefit from in the enterprise environments. Just like any other transformation in Warehouse Builder, data quality transformations are accessible in scripting language and through public application programming interfaces (APIs), in addition to the graphical user interface. The audit data pertaining to the execution of data quality processes is available in the same place and format as the audit data about other ETL processes.

The integration of data quality functionality in Warehouse Builder is not just for the sake of convenience. The higher goal is to promote the disciplined approach to data quality, encouraging the ETL developers to think of data quality as they design the data integration processes and enabling them to incorporate it into these processes, as opposed to treating it as an afterthought.

DATA QUALITY IN ORACLE WAREHOUSE BUILDER

Since Warehouse Builder 10g Release 2 was released, Warehouse Builder provides a complete and integrated data quality solution in the Oracle database. Based on a single metadata repository, Warehouse Builder combines data profiling capabilities, data rules and data auditors with robust data modeling, data integration and the more classic data quality solutions such as name and address cleansing and fuzzy matching and merging.

The latter two are data quality features, but are really most often used in the data integration space. For this paper we are obviously focusing on the matching and merging of data.

Matching and Merging of Data

Matching is the process of determining, through business rules, which records refer to the same logical data. Merging is the business rules-driven consolidation of the data from the matched set into a single record.

There are multiple uses of match-merge for different purposes that have their own terms:

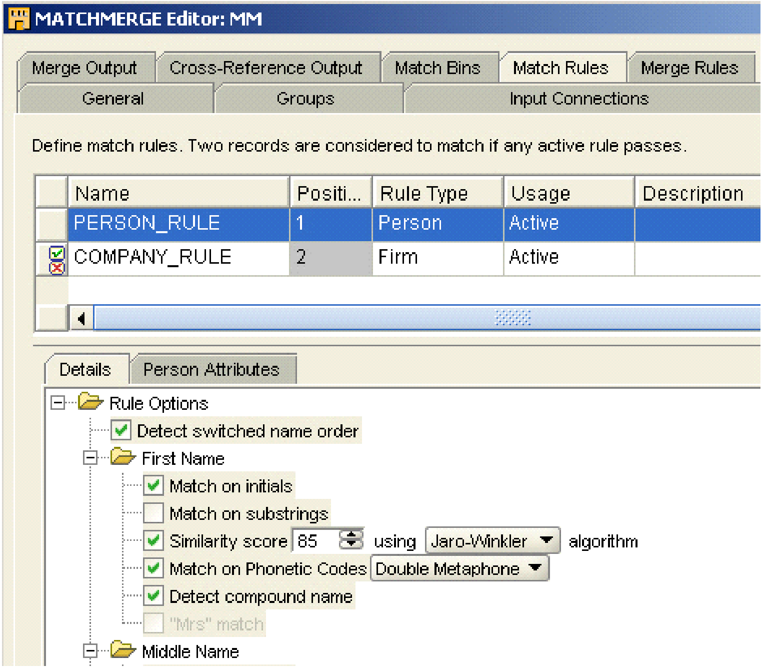
- Deduplication is the process of matching and merging for the purpose of removing the duplicate records, especially customer-related records. This contributes to achieving the single view of the customer.
- House holding is the process of matching customers belonging to the same household, usually identified by the same address. Customer names are not merged; however, they are linked to the address that is stored once. The benefit of house holding is the improved ability to understand and target customers.
- Record linking is the more generic instance of house holding. The records may need to be linked for purposes other than determining households, for example, linking business branches and subsidiaries to one parent entity.

It is also helpful to put the various terms pertaining to matching into perspective with Warehouse Builder. Warehouse Builder employs the elements of *fuzzy logic* and provides both *deterministic* and *probabilistic* matching algorithms:

- In general, fuzzy logic resembles human reasoning in its use of approximate information and uncertainty to generate decisions. In relation to matching, the term is used loosely to describe the approach that relies on rules that are imprecise rather than precise and operates on data with boundaries that are not sharply defined.

- Deterministic matching gives equal weight to the different types of information a record may contain. For example, a deterministic approach might place equal reliance on a match between the names on two records or a match between two birth dates.
- Probabilistic matching exploits the statistical probability that a match on particular items is more or less likely to indicate that the records are the same. For example, birth date information is subject to errors made by a mistake on a single digit, and the number of possible birth dates is relatively small. Names, in contrast, are more likely to be recognizable even if a single error is made. Probabilistic matching thus allows assigning appropriate weights to different attributes and then compares the total score to the threshold that defines a successful match.

Warehouse Builder accomplishes these tasks by using the match-merge operator with a wizard-driven interface. The match-merge operator comes with a powerful set of UI-controlled matching and merging algorithms. The user simply chooses the rule appropriate for the data being matched and the business requirement, followed by selection of parameters that control exact behavior of the rule. The whole operation is done entirely graphically with checkboxes, drop-down boxes, and scroll bars. It is possible to create very sophisticated match rules with only a mouse, i.e., not typing in any text. An example of a match rule creation is shown in Figure 6.



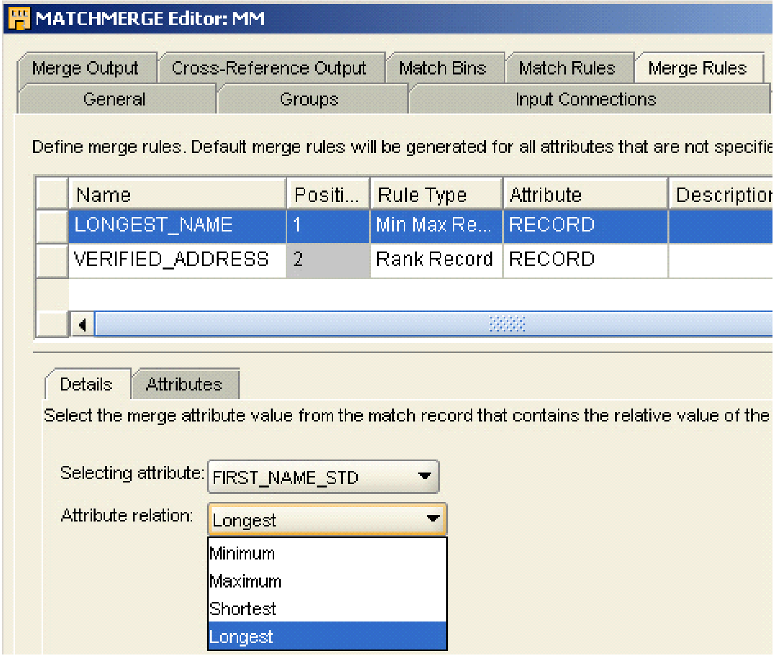
Define match rules. Two records are considered to match if any active rule passes.

Name	Positi...	Rule Type	Usage	Description
PERSON_RULE	1	Person	Active	
COMPANY_RULE	2	Firm	Active	

Rule Options
☑ Detect switched name order
First Name
☑ Match on initials
□ Match on substrings
☑ Similarity score 85 using Jaro-Winkler algorithm
☑ Match on Phonetic Codes Double Metaphone
☑ Detect compound name
□ "Mrs" match
Middle Name

Figure 6. Creating a Match Rule in Match-Merge Operator

Similar to match rules, selecting the rule types and then supplying the parameters applicable to each rule create the merge rules. Figure 7 shows an example of how a merge rule is created.



Define merge rules. Default merge rules will be generated for all attributes that are not specifie

Name	Positi...	Rule Type	Attribute	Descriptior
LONGEST_NAME	1	Min Max Re...	RECORD	
VERIFIED_ADDRESS	2	Rank Record	RECORD	

Select the merge attribute value from the match record that contains the relative value of the

Selecting attribute: FIRST_NAME_STD

Attribute relation: Longest

Figure 7. Creating a Merge Rule in Match-Merge Operator

With all the ease of use a graphical interface provides, the flexibility is not sacrificed. Match and merge rules both have custom rule types, allowing for a PL/SQL-style procedural logic to be put in. Multiple rules can be called from other rules because they are represented as functions with input parameters.

At runtime Warehouse Builder executes the match-merge rules and transforms the input data, potentially containing duplicates and variations, into the consolidated output. Note that the implementation of match-merge operator in Warehouse Builder is entirely Oracle technology, with no reliance on a third-party.

In match-merge functionality, the following is new in the Paris release:

- More accurate and better performing phonetical algorithm—“Double Metaphone”
- More accurate and better performing similarity algorithm—“Jaro-Winkler”
- Optimization of matching “incoming” records against “existing” records through a “Match New Records Only” option, resulting in fewer matching comparisons performed

A small use case

One of those hard to solve problems in today’s consolidating world is matching data elements from disparate sources to create a master reference for your corporate data.

This short example shows Oracle Warehouse Builder's match/merge capabilities in a fairly common scenario. We will de-duplicate customer data from three different systems using the match/merge operator.

The power of the match/merge operator is that it is rule based and makes use of powerful algorithms rather than of hand coded custom routines. As part of the core product, match/merge is one of the most powerful tools in your toolbox to resolve this common problem.

This small glossary explains some terms up-front and can act as a reference later on. It also has a definition of how the term is used in Warehouse Builder in this specific case.

Term	Definition
Match	The first step in match/merge. Match uses match rules leveraging algorithms to detect potential matches. Many match rules can be combined. Sets of attributes (or individual ones) can carry individual rules
Merge	Using rules to actually apply a merge on the potential matches. Each attribute can carry one or more individual merge rules.
Bin	The most important concept for performance of the matching. A bin groups or partitions (not the DB feature!) data and matching only occurs within the bin.
Match or Merge Rules	The individual algorithms applied to a set of attributes including the thresholds etc constitutes a match or merge rule. Rules can be composite via a custom rule.

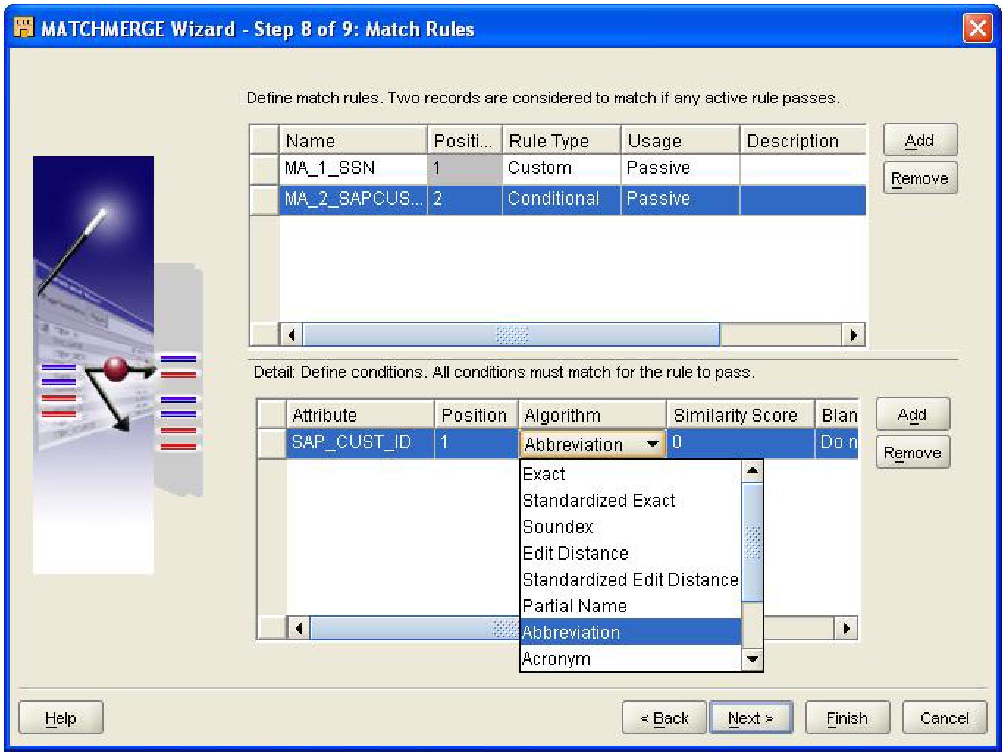
Match Rules

We will implement a complex rule scheme using the following rules:

Data Element	Rule Description
SSN	If the SSN is not null and not equal to 999-99-9999 then use fuzzy (edit distance) matching
SAP_CUST_ID	If the SAP_CUST_ID is not null then use partial (abbreviation) matching
XYZ_CUST_ID	If the XYZ_CUST_ID is not null then use exact matching

Data Element	Rule Description
ABC_CUST_ID	If the XYZ_CUST_ID is not null then use exact matching
NAME (composite)	If first name and the last name are not null then use fuzzy (soundex) matching

A completed set looks likes this:



MATCHMERGE Wizard - Step 8 of 9: Match Rules

Define match rules. Two records are considered to match if any active rule passes.

Name	Positi...	Rule Type	Usage	Description
MA_1_SSN	1	Custom	Passive	
MA_2_SAPCUS...	2	Conditional	Passive	

Detail: Define conditions. All conditions must match for the rule to pass.

Attribute	Position	Algorithm	Similarity Score	Blan
SAP_CUST_ID	1	Abbreviation	0	Do n

Merge Rules

The merge rules (determining how we choose each attribute) are based on the following:

Data Element	Rule Description
NAME_M	The longest non-null middle name
SSN	The most common SSN
NAME_F	The longest non-null first name from Table A
CUST_SEQ	From the same record as the merged SSN
SAP_CUST_ID	The most common SAP_CUST_ID with 7 characters

Rather than creating all this in the UI we will choose scripting in OWB to create these rules in the metadata. A short overview of the commands is as follows:

```
#Creating a merge rule set
#Rule 1:
#Note that the order of the last two settings has to be exactly like
#this! TYPE before Attribute.

OMBALTER MAPPING 'CUST_MATCH_MERGE' \
    ADD MERGE_RULES 'ME_1_NAMEM' \
        OF OPERATOR 'MATCHMERGE' SET PROPERTIES (TYPE) VALUES
('MM_MIN_MAX') \
    MODIFY MERGE_RULES 'ME_1_NAMEM' \
        OF OPERATOR 'MATCHMERGE' SET PROPERTIES (ATTRIBUTE_NAME) VALUES
('NAME_M') \
    MODIFY MERGE_RULES 'ME_1_NAMEM' \
        OF OPERATOR 'MATCHMERGE' SET PROPERTIES (MIN_MAX_TYPE) VALUES
('MM_LONGEST') \
    MODIFY MERGE_RULES 'ME_1_NAMEM' \
        OF OPERATOR 'MATCHMERGE' SET PROPERTIES (MIN_MAX_ATTRIBUTE)
VALUES ('NAME_M')
```

This first rule creates a pre-built type rule picking values for NAME_M (middle name) via the longest of the middle name fields in the matched records.

The following is an example of a more complex rule:

```
#Rule 2:
#Custom rule text is added to a variable and called from OMB.
#Note the escape characters in the PL/SQL text:
# Double quotes are escaped by a slash
# Single quotes are escaped by a single quote

set custom_rule2 "fName varchar2(2000) := null;
BEGIN
    -- return the longest first name from table a
    -- in table a, CUST_SEQ is not null
  FOR i IN M_MATCHES.FIRST .. M_MATCHES.LAST LOOP
      IF  M_MATCHES(i).\"NAME_F\" IS NOT NULL and
          M_MATCHES(i).\"CUST_SEQ\"  is not null THEN
        IF fName IS NULL OR LENGTH(RTRIM(M_MATCHES(i).\"NAME_F\")) >
LENGTH(RTRIM(fName)) THEN
          fName := M_MATCHES(i).\"NAME_F\";
        END IF;
      END IF;
  END LOOP;
  RETURN fName;
END;"

OMBALTER MAPPING 'CUST_MATCH_MERGE' \
    ADD MERGE_RULES 'ME_2_NAMEF' \
        OF OPERATOR 'MATCHMERGE' SET PROPERTIES (TYPE) VALUES
('MM_CUSTOM') \
    MODIFY MERGE_RULES 'ME_2_NAMEF' \
        OF OPERATOR 'MATCHMERGE' SET PROPERTIES (ATTRIBUTE_NAME) VALUES
('NAME_F') \
    MODIFY MERGE_RULES 'ME_2_NAMEF' \
        OF OPERATOR 'MATCHMERGE' SET PROPERTIES (CUSTOM_TEXT) VALUES
('$custom_rule2')
```

For more on the scripting on the merge rules have a look at the Warehouse Builder blog entry here.

Match Merge is one of the most powerful features in Oracle Warehouse Builder but for some reason rarely used. This session and paper aims to change this.

CONCLUSION

We have discussed the concepts of data quality, their importance in an enterprise as well as the distinct advantages of performing data quality processes inside a robust data integration product—Oracle Warehouse Builder.

Oracle Warehouse Builder addresses the challenges of data quality by offering data assessment, data cleansing, data integration, and data monitoring in one tool. It promotes the disciplined approach to data quality by making data quality processes easily available in the same development environment as the regular data transformation processes.

Warehouse Builder offers the flexibility and accuracy that data integration projects demand, yet at a low cost. Matching and merging is often the biggest issue in these projects, and Warehouse Builder provides a solution embedded in the database delivering superior performance at a low price point.

By implementing data integration with Warehouse Builder, you will build a solid foundation for data quality in your enterprise and make sure that matching and merging of your data is a real traceable science. We would not want to rely on black magic when the auditor comes by and questions your data points.

Matching and merging data – Black Art or Exact Science
January 2008
Author: Jean-Pierre Dijcks

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com