Warehouse Builder 11*g*

# Best Practices for a Data Quality Process with OWB

*May 2008*

ORACLE

# Best Practices for a Data Quality Process with OWB

## INTRODUCTION

Maybe the introduction to this paper should be the following paragraph, what the paper is not intended to do, but that does not quite make sense. As the title says, the paper focuses on how to implement a data quality process with Oracle Warehouse Builder 11g Release 1 and the best practices associated with such an implementation in a data warehouse context.

There are many things that must and should be done around data quality, but some are beyond the reach of what we are describing here. We will describe an architecture to ensure a certain quality level in a target system, like a data warehouse or a master data management application. We will not attempt to solve any and all data quality issue in the organization (like data entry issues).

So based on the above, you can expect to find a reference model for a data quality architecture using the Oracle Warehouse Builder tool set. As much as possible this will be supported with real examples.

## WHAT THIS PAPER IS NOT

As we said in the introduction, this paper is not pretending to solve all of your data quality issues in the entire organization. The paper focuses on the recipient systems of any data, then discusses how to keep these systems at the appropriate quality level.

If you read the literature on quality management you will discover a lot of discussions around symptoms and root causes. The goal in total quality management is to tackle the root cause and not address just the symptoms. Here we are looking at addressing the symptoms and hopefully start to address the root cause by documenting the potential root causes. The reason for doing this in this way is to ensure your data warehouse or MDM project will actually finish at some point in time.

Consider the following, you receive bad data due to missing address elements. The root cause in the end appears to be that the data entry personnel never fills out these details due to timing issues with the ERP system. While you are doing the data warehouse, the only thing you can do is measure the defects and report them. If you get side tracked in trying to solve the root cause, you need to change the ERP system; change the mindset of the personnel and then hope it is now fixed. While this sounds like a good idea, it is completely outside of your project scope, it does not make your project go faster, and ultimately if you fail to deliver your project, it all sounds like an excuse…

So here we are going to focus on architecting a solution for data quality that hopefully will influence (once the system is up and running) the downstream data owners. This architecture will however still allow you to finish your project and to set the expectations of the data quality beforehand so your project becomes a success.

Oh, and one other thing. We are not going to describe in detail what all the components are for this system. The paper will assume a basic understanding of for example data profiling and what a data profiling tool does.

### What is a target system?

Since we are going to discuss a data quality architecture for a target system, here is what we will be discussing.

First of we will look at the data warehouse and specify how a system should be set up to create appropriate quality levels in the data warehouse. We will discuss what components should be used when and where and how to install and configure.

Secondly we will discuss a master data management system and how to handle data quality such a system. Here we will introduce some new product components and discuss the set up for this system.

## DATA QUALITY IN A DATA WAREHOUSE

We are going to from the ground up here. First we introduce a basic architecture that applies to most every data warehouse. Next we start to place the data quality components into this architecture and lastly we go into the details of these components as to what they do and how to utilize and set them up in an efficient manner.

### A Basic Data Warehouse Architecture

I'm sure there are more in-depth discussions possible about this architecture, but for the purpose of this paper we want to set a baseline as to what components make up a data warehouse.

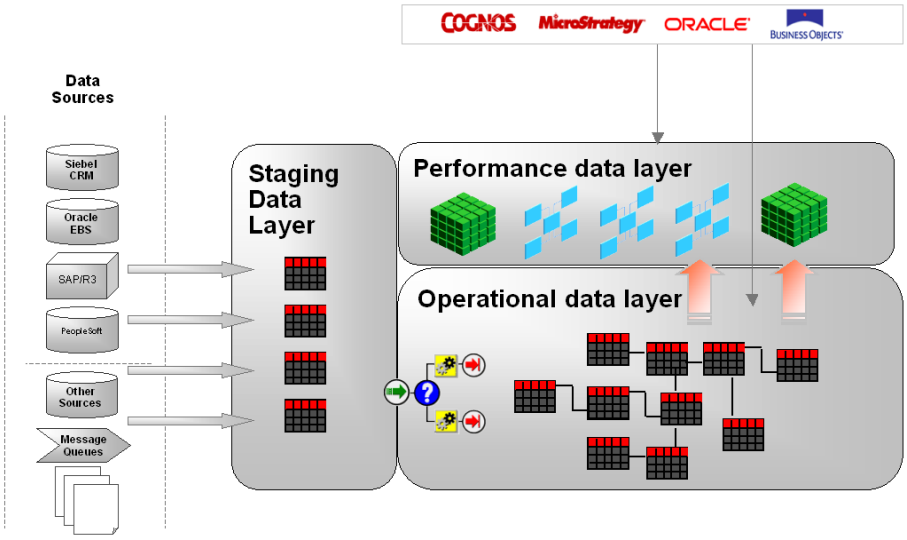


**Figure 1 The basic data warehouse architecture**

In above architecture you will see the classic model of a staging area, a data store with operational data in 3NF-style schemas and a performance layer to enhance performance of certain query applications.

ETL or data integration processes run between the source and the staging layer, run between the staging layer and the operational data store and potentially run between the operational layer and the performance layer.

Both the performance layer and the operational data layer are accessible for queries, although they may service separate user communities. The performance data layer is comparable to a data mart, but resides within the same database instance. By doing this, performance for data movement is a lot better than when you move data outside the instance.

In an Oracle database, Oracle Real Application Clusters (RAC) is the basis for this architecture as RAC allows you to provision the resources to the appropriate

applications. ETL could be run on one node, the query layers could be run on the remaining nodes. The nice thing here is of course that when no ETL is done, that node becomes available for other activities. For example this node could be used for data profiling, or for example additional resources for the query applications.

## The Data Quality Architecture – A Firewall

Using the above described data warehouse architecture we will start to fit the quality pieces into it.

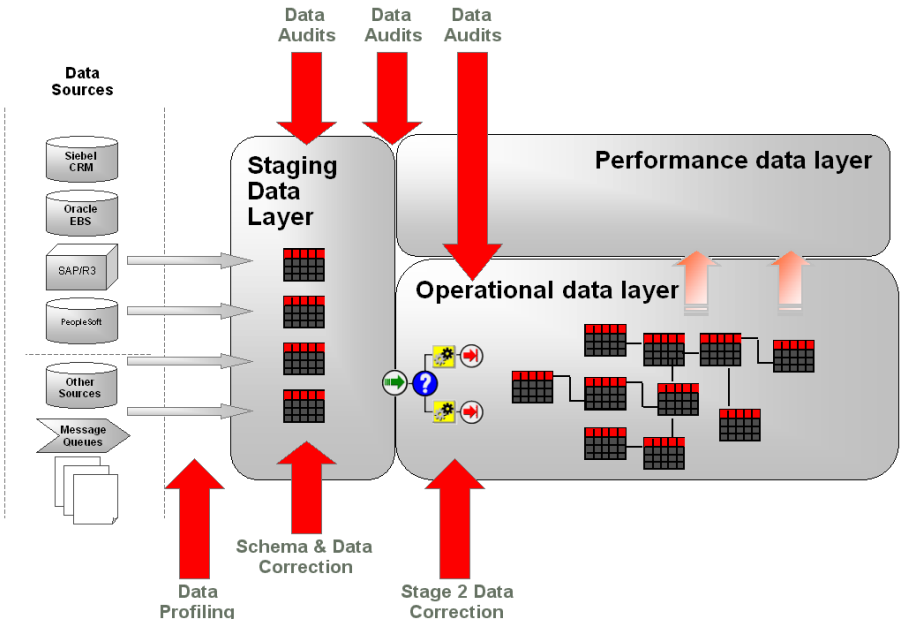


**Figure 2 Data quality in the data warehouse architecture**

There are multiple aspects to the data quality firewall architecture. In the above architecture they are placed in the position where they are applied. We will now briefly define the components, each components purpose and where to apply the components to get the maximum value out of the data quality firewall.

### Data Profiling

The famous cousin in data quality land is the first to be addressed as it is always on the forefront of any data quality discussion. It is also the first step in a quality process as data profiling allows you to discover and describe your data based on the data alone.

If you look at Figure 2 you will see that the data profiling arrow points to the actual data flow between source and staging areas. This is because in actual fact you typically profile in neither of these two. With Warehouse Builder you can profile the source, however because the profiling algorithms are processing intensive this might strain the operational system too much. So in most cases you move data somewhere else and profile it. The staging area is probably not that place, because

by definition it captures the changes between the last run and today's data set. Where do we profile then?

As with most profiling tools, Warehouse Builder creates a separate place to do the profiling. The difference is that Warehouse Builder does all the hard work in the database without moving data out of that secure area. If you check the box that states to move the data, Warehouse Builder moves the data (transparent to the user) into a profile workspace, then profiles the data and stores the results in that profile workspace.

The best practice is to place this profile workspace on a resource heavy environment like the data warehouse to ensure appropriate performance for the data profiling task.

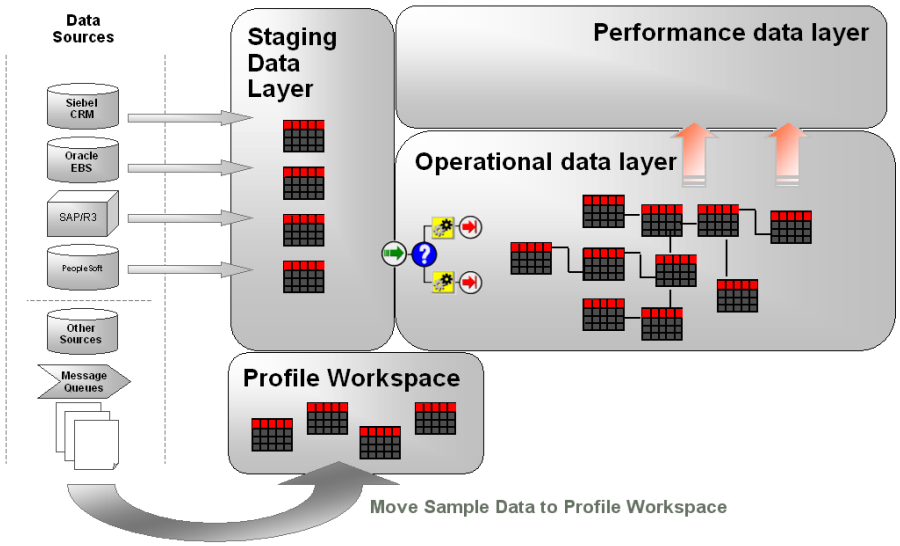


**Figure 3 Adding the profile workspace into the architecture**

Even on a resource heavy machine, always start out with a sample set of data. Within Warehouse Builder you can choose the sample size easily for your data.

Even with the sample size at a small sample, you need to remember that a lot of the profiling resources are spent on working on column relationships. So again, don't just profile every column and try to see if there are relationships between all of them, because you are not efficiently using the resources.

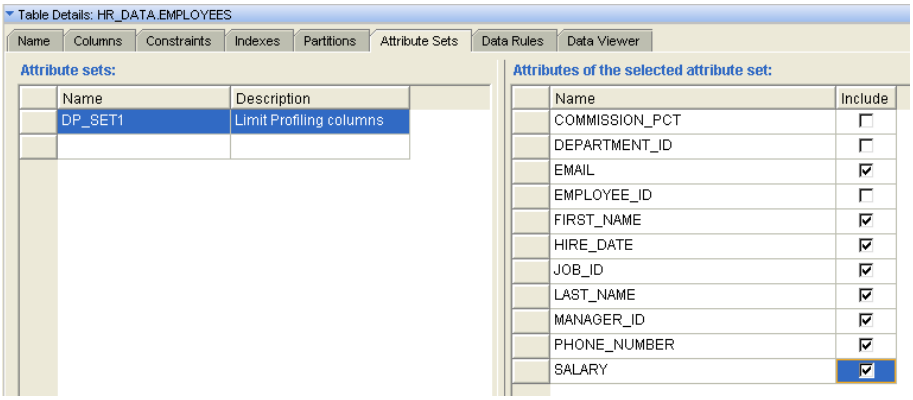


**Figure 4 Applying an attribute set to the source table**

Use attribute sets in Warehouse Builder to determine which columns on a table are candidates for profiling, then in the profiling editor choose which ones you really are going to profile. For example, do you really want to profile a system generated numerical key field out of the ERP system or use those resources to profile that varchar2(4000) column labeled "additional information"?

### Data Rules

If data profiling is the much hyped cousin, data rules is the quiet one in the back that does all the hard work. Without data rules, there would not be a solution as we described it in the architecture. The data rules are the one component that allows Warehouse Builder to create a unique architecture leveraging the definitions for quality across the system.

A data rule is a separate metadata object storing the business rules in a actionable format. Rules can be for example SQL or PL/SQL or can be written in regular expressions if you are describing patterns. Warehouse Builder provides built in examples and in the case of Oracle Data Watch and Repair (see next chapter) provides pre-built rules for specific MDM applications.

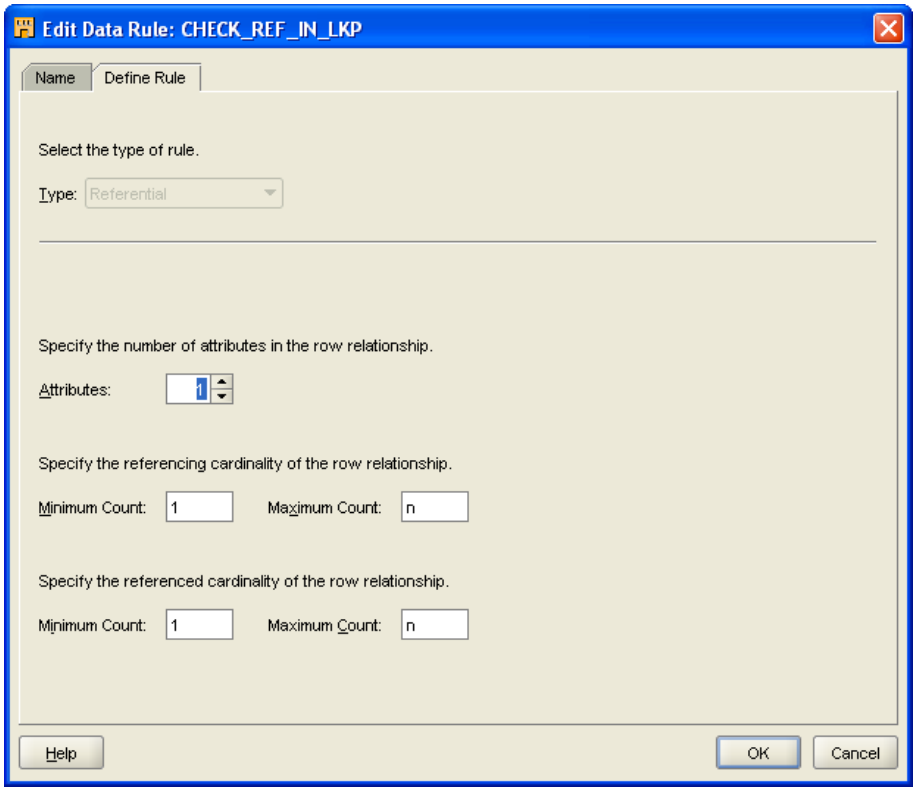


**Figure 5 A referential data integrity rule (not a constraint)**

By creating rules as metadata objects in the Warehouse Builder repository you can use these rule to do a number of very powerful things:

- Create correction mappings to correct data issues by applying the rule to the table in ETL
- Create corrected schema definitions by applying the profiling results and the data rules to the table
- Measure the rule compliance without moving data

In the next two paragraphs we will discuss these topics as they are key in our architecture and in our best practices.

For data rules themselves, make sure you create them as these separate entities in the Warehouse Builder repository. Make sure they are shared in the public data rules folder is they are to be used across projects. And most of all, just make sure you derive them or create them! This is the corner stone of implementing the above best practices.

#### Schema and Data Corrections

If we go back to Figure 2 and look at where we are doing the corrections it is important to realize these corrections are both done within the warehouse database.

For data and schema corrections that will change data do this from source to staging area (if you are loading files use external tables). Apply the data rule to the source table, then generate the correction mapping into a corrected table. Place both the correction mapping and the corrected table in the staging area and mark this table in some way as generated.

This new table will have check constraints (if you have domain rules) on them for example or be split out into two tables when there is functional dependencies detected, but will also have a mapping that will convert faulty data to correct data.

The reason for doing this from source to target is that you now load the data, correct it etc in a single mapping. This mapping does a staging step in it (you will have this table materialized in the staging schema) and then loads into the corrected table. Anything that cannot be corrected is stored in an error table that is associated with the staging table. So you still have the simple extract into the staging table and you do the hard work in Oracle. The good thing is that this is all generated. So when the data rules changes, or the data coming from the source, you can re-generate everything.

If you want to just generate a corrected staging table based on the profiling data types found (be careful with this, it is a staging area) you can do this by ignoring the correction on the rules. Then the mapping generated is just a simple load mapping with no data corrections, just loading source into corrected schema. This way you can get the entire staging area generated as long as there is at least on data rule on the source tables.

In this step you should only do simple corrections. For example the domain value string matching. If you start doing things like referential integrity you should do these cleansing steps after loading the staging area. You will need all data in the operational store to make sure the lookups are valid. That is why there is a specific single data correction step when moving into the operational area.

That stage 2 data correction step is to do the real heavy lifting. You should always do this second stage within a single instance to remove all potential performance bottlenecks from the equation. Here you would also start doing the match/merge and name and address cleansing and standardization steps. The resulting data is clean and standardized; it is validated and correct according to all data rules.

#### Data Auditors

The last section covers data auditors. Again data auditors are based on the data rules, but rather than correct schema definitions or data elements a data auditor just measures the compliance of the data to the data rule.

There are three places to measure and document data quality:

1. Raw incoming data in the staging area
2. After initial cleansing

3. After final cleansing

If you measure it this way, you will get an accurate measurement throughout the system. Especially the first two measuring points are interesting as they tell you the raw data quality compared with automated cleansing. As a first step this is something that every system should do. The third phase is a bit more advanced.

There is one more reason to measure in either point 1 or point 2. If you measure there using the data auditors in a process flow, you can verify incoming data quality and decide to not load this data into your warehouse because it does not meet the threshold set of the data by the business.

This decision point is something you should value. Rather than the above where you load bad data, now need to get it back out, you make an informed decision whether or not to load the data.

#### Data Quality Firewall

With all of the above in place you can now assure the end users that you have a firewall in place allowing you to cleanse your data, protect the system from incorrect data being loaded and being able to report on the quality indexes on the data warehouse.

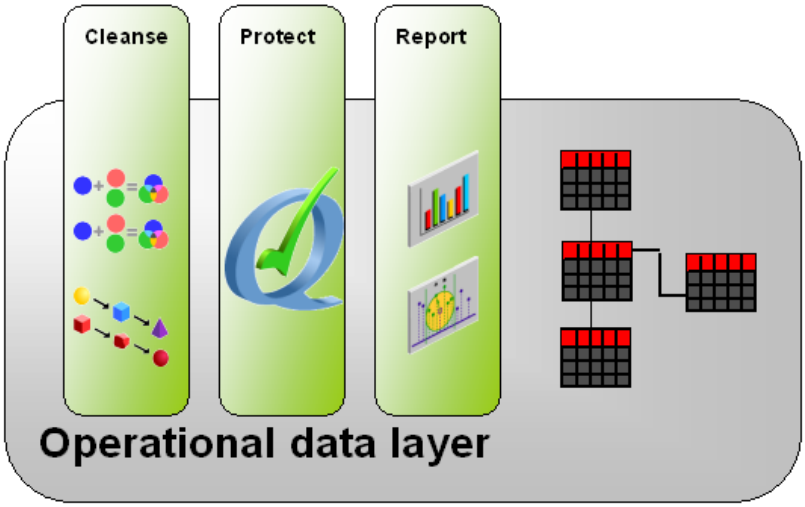


**Figure 6 Three pillars of a data quality firewall**

### Feeding External Systems

The main goal of data quality is to improve quality. Obviously. However this is not something simple and cheap in most cases. Point being that you don't want to duplicate effort.

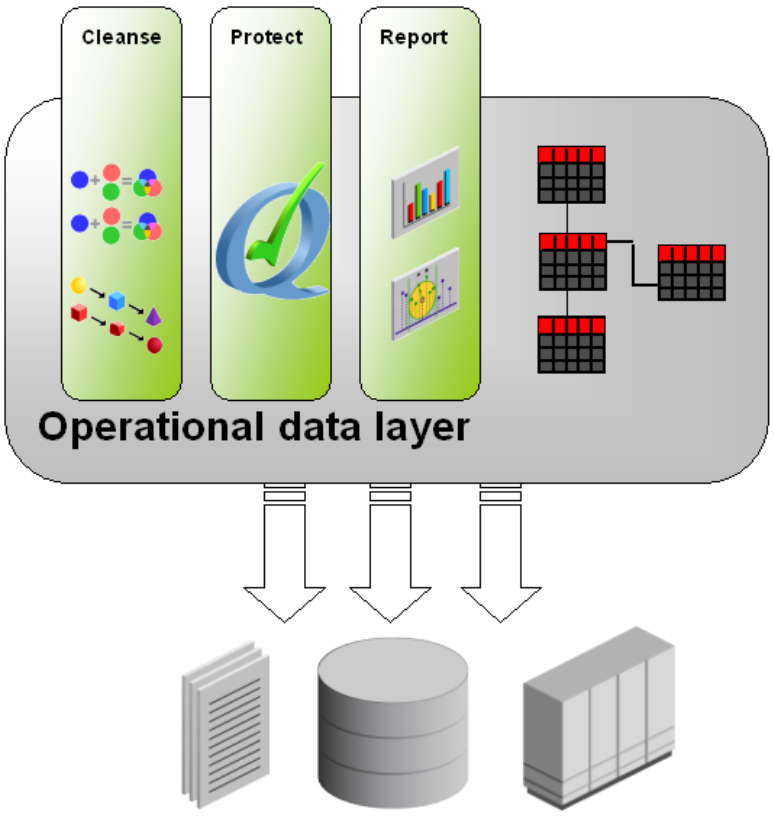


**Figure 7 Load all systems from the cleansed source**

If you have a data warehouse like discussed above, than rather than doing the cleansing actions again, leverage this clean data to feed other systems. Obviously the opposite is true as well. If you have a master data repository, make sure you load the appropriate data warehouse components from the master data repository.

The next chapter deals with how Oracle Master Data Management delivers data quality on top of Warehouse Builder for their customers. If you are looking at both solution, make sure to decide which application will take the lead in data quality for which data elements.

## DATA QUALITY IN A MASTER DATA MANAGEMENT SYSTEM

As a best practice for quality data in the MDM solutions from Oracle, a new solution is created on top of the Warehouse Builder components.

### Oracle Data Watch and Repair for Master Data Management

Oracle Data Watch and Repair for Master Data Management (DWR) is an out-of-the-box profiling and correction solution created to assist data governance processes in Oracle's Master Data Management (MDM) solutions.

#### Why is this Important?

MDM applications must successfully consolidate and clean up a system's master data, sharing it with multiple connected entities to achieve a single view of the data. However, MDM systems face a never-ending challenge: the constant state of flux of master data.

Not only does master data quickly become out of date as new events happen and need to be captured in the system, but also any incoming data can potentially be inaccurate, either from entry mistakes or purposely misrepresented data.

Therefore, in order to leverage this achieved single view, it is crucial to implement a data governance plan. Being able to look at the data constantly, thoroughly and easily ensures that any data decay can quickly be noticed and the necessary correction or cleansing steps can be taken.

Consequently, it will be easier to keep up reliable and useful data to make asserted and timely business decisions. Data Watch and Repair is a tool created for these specific tasks.

#### What is in the Solution?

The product, therefore, includes the following,

- **Data Profiling –** to discover the structure of the data and understand it
- **Application of data rules –** to evaluate compliance and quality of the data
- **Correction mappings –** to specify the necessary cleansing strategy to be used with data not compliant to a given data rule

In addition to these capabilities, the product also comes with a set of pre-written data rules that are common and relevant in an MDM context. Sample data rules for both customer and product hubs are created for out-of-the-box usage to perform basic data governance tasks. Moreover, they serve as example rules that illustrate how to define data rules, facilitating the learning of how to implement new data rules in the rule syntax.

Being in charge of data lifecycle activities, data stewards are most likely to be the primary users of this product. A typical day as a data steward might involve running

initial data profiling tasks to inspect the data and to run data fixing routines if necessary. As part of these activities, a data steward could also customize the data rules to gain more insight on suspect data or as a consequence of a new data quality bar imposed or changing business requirements.

### Solution Architecture

The solution is completely based on the Warehouse Builder data quality solution but is tuned to work in concert with the MDM solutions. For example specific rules are governing the write back of corrected data to the MDM system. Those rules are implemented using Warehouse Builder in the DWR solution.

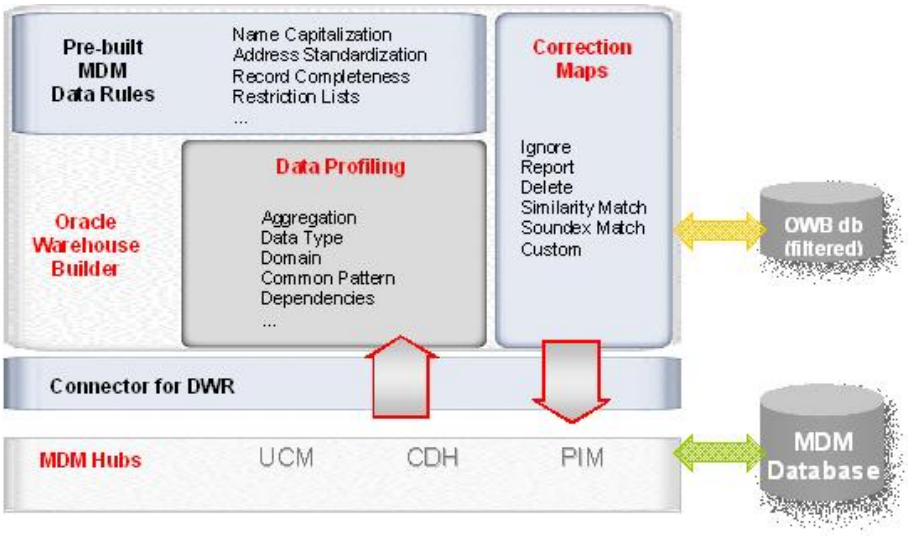


**Figure 8 Data watch and repair architecture overview**

To comply with the MDM solution rules for writing back corrected data, the mappings that do the write-back are designed at Oracle as well. This allows you to utilize the correct mechanisms for integration the corrected data in the MDM solutions.

### Pre-Defined Data Rules

In addition to the connectors, MDM-specific data quality metrics are addressed by each option. Some are met with the built-in functionality in the profiling step, while others have been addressed with custom data rules. Data rules addressing additional requirements for Customer Hubs have been included. CDH and UCM share the same rules as they both master customer data. The data quality metrics addressed for CDH and UCM are shown in the table below.

**Table 1 Pre-defined rules for the MDM solutions**

| DQ Metric | Customer Rule | Description or Example |
|---|---|---|
| ***Completeness*** | Attribute completeness | Discovers # and % of null values |
| | Contact Completeness | Requires all name, address, SSN, … are not null |
| ***Conformity*** | Data Type | Data type, length, & precision documented & found |
| | Data Domain | All values are within specified domain |
| | Restricted values | Values are not in a list of restricted values: e.g. "66666…"… |
| ***Uniqueness*** | Unique Key discovery | Discovers # and % of unique values |
| ***Pattern and Common Format*** | Name Standardization | First, Middle and Last name not null & properly capitalized |
| | Common Pattern | Common pattern required (e.g. phone, email), % conformity |
| | Name Capitalization | "Aaaa", "Aaa Bbbb", "Aa-Bbb", … |
| | Extended Phone Numbers | More extensive definition of allowable phone number formats. Can extend to specific country formats, etc. |
| | International Phone Numbers | |
| ***No Access Lists*** | by Name Only | Restriction lists for specific records, filtered by name, SSN and/or emails |
| | by Name or SSN | |
| | by Email List | |

With these pre-defined rules the time to solution is much shorter than without the rules. If you were to implement these rules you would have to be both an expert in the MDM solution and an expert in Warehouse Builder.

Because the rules are part of the Warehouse Builder based solution, you can always customize the rules and regenerate the correction mappings. Because this is driven from the data rules (see also the previous chapter and its discussion on these rules) you need a much shorter time to update anything derived from these rules.

## SUMMARY

The above text should give you pointers how to effectively leverage the Warehouse Builder components in two areas. One is the data warehouse where data quality is becoming a bigger topic every day. The other is in the Master Data Management arena, where the system cannot live without appropriate data quality.

For the data warehouse, make sure to leverage the data profiling to discover issues or verify the content of data elements. Implement data rules and drive the remaining solution elements from these data rules. Data and schema corrections, as well as data auditors allow you to improve your data quality in a measured way. Use these measurements as feedback to the business data owners to improve the root causes.

In Master Data Management, start with the same solution based on Warehouse Builder and leverage the content of the pre-defined data rules to quickly start the data quality effort on your MDM system.

In summary, Oracle Warehouse Builder delivers a high-value data quality solution for data intensive applications like MDM and your data warehouse. If used in the correct architecture it is a sure thing to add value to the business by improving data quality for the business.

**Best Practices for a Data Quality Process with OWB**
**May 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**