



Fixing Data Quality in Data Warehousing Projects

This whitepaper discusses in depth on what and how to fix your data quality issues in your Data Warehouse projects.

A White Paper By:

Subra Suppiah & Christopher Waters
Copyright © Knowledge Discovery (M) Sdn Bhd
. All Rights Reserved. 2008

www.kbase.com

Contents

1.	Introduction and Purpose.....	5
1.1	Credits and Acknowledgments.....	5
2.	The Politics and the Concerns	6
3.	Project Scoping - Data Quality Considerations :.....	8
4.	Data Quality Problems:	10
5.	Data Cleansing Considerations - Manual Orientation.....	12
6.	Error Handling and Reporting	15
7.	Data Quality Improvement & Resolution.....	19
8.	Mechanical based validation during Data Migratio	24
9.	APPENDIX A - Data Quality Scoping Questions	26
10.	APPENDIX B - Data Quality Problem Errors.....	29
11.	APPENDIX C - References	33

Preface

Building a data warehouse is often a substantial investment for an organization. This substantial investment causes organizations to view the data warehouse as an asset of the organization. Justifying substantial asset purchases often requires a business case with estimated return on investment. Accordingly a major effort occurs during planning phases for developing an estimated return on investment. The business case foundation is providing the ability for answering business questions, which would not be answerable in the absence of a data warehouse.

Given a set of business questions, which estimate data warehouse return on investment, a significant basis for business question value is data quality. Providing accurate data to the business questions enables realizing full value, while lack of data accuracy disables realizing return on investment. Therefore the impact of data quality, or lack thereof, significantly impacts data warehouse return on investment.

Should an organization seek improvements to data quality and those improvements are necessary to the success of the data warehouse project, those improvements will only come at a price. The price of improving data quality can be significant. Applying transformations and corrections to operational data is the process of improving data on an “after the fact basis” and subjecting data warehouse deployment engineers to the inherent difficulties incurred by the reverse engineering process.

Left unattended to, data quality may become a significant cost factor of delivering the data warehouse, or when not addressed, may become a significant factor detracting from realization of return on investment. Upon recognizing the significance of this issue, organizations seek an understanding of data quality during data warehouse planning phases. Accordingly, understanding and measuring data quality becomes a necessary goal of data warehouse design engineers.

Seeking an understanding of data quality, the data warehouse planner then sets about looking into operational systems, intending to understand and measure data quality. At this point, the data warehouse design engineer then faces the challenge of accessing operational data in a relatively unrestricted manner. The challenge for accessing operational data is, at least, identical to the challenge facing the organization’s business users. In addition, the data warehouse design engineer often undertakes an added challenge for designing a multiple subject data repository, organizing and capturing historic data. Accordingly data quality needs inspection within single subjects, then across multiple subjects, and over current and historic time periods. As a design engineer, the data warehouse planner faces the need for a data warehouse for a comprehensively assessing data quality.

The need for having a data warehouse, prior to deploying a data warehouse lends the notion of the “chicken before the egg syndrome.” When one considers which came first, the chicken or the egg, one recognizes chickens hatch from eggs. Likewise, when considering data warehouses and data quality, the data warehouses is the tool for understanding data quality. In that the data warehouse is the platform providing cross

functional analytical capability, comprehensive analysis of data quality first requires deploying a data warehouse.

Unlike the eternal “chicken or egg” question, the issue of data quality is practically addressable. This paper offers two realistic approaches. The first approach promotes assessing data quality, deriving results from interviewing data custodians, who own the data itself. This interview approach perches on the knowledge and opinions of those who are most familiar with the data. As a result of this approach, the successes of the data warehouse project becomes dependent upon the knowledge of these people and the interview skills of the design engineers. The foundation of the second approach relies upon real operational data being loaded into data staging areas. These staging areas take on the operational characteristics and benefits of an operational data store environment. Within this environment, the engineering staff assesses and measures actual data content. Using the enabling characteristics of the data warehouse platform itself, the data warehouse platform participates in the design process. Practically applied, the second approach yields real results, supported by real data.

Developing a perspective on data quality and addressing the issue is the responsibility of the data warehouse design engineering staff. Successfully addressing data quality issues sets achievable client expectations while promoting the value of the data warehouse project itself. Certainly the issues are less than simple, but with reasonable application of wisdom, judgment, and common sense, experienced data warehouse practitioners plan, design, and implement methods for measuring, monitoring, and demonstrating data quality.

Given the validity of the business case, which justifies building a data warehouse, measuring data quality in the absence of a data warehouse may lead to unpredictable cost modeling, haphazard realization of return on investment, or both. Therefore a properly funding data quality assessment and applying scientific approaches to determining data quality greatly contributes to a successful data warehouse project.

1. Introduction and Purpose

This document provides those involved in the Data Warehouse (DW) implementation process with a consolidated view of core data quality issues, considerations, activities, and sample templates to document the effort. Failing pursuit of data quality during the DW project invariably creates requirement for addressing the issues when you have little choice and time (i.e. The chicken before the egg syndrome). If the focus emphasizes developing an early data quality measurement versus waiting to a later stage, obtaining strategic insight, obtaining a headstart and funding for overcoming data quality challenges.

This document was not intended to be an all encompassing document due to wealth of material being available in the market place and on the Internet. This paper will then focus in more detail on the most significant items. A consolidated view of most of the activities one should consider during a DW project. This document is, in present state, more oriented toward external usage's by a practitioner, hence the cut and pasting appearance of topics verses the fluidity one achieves while writing a formal or external document or book. Those clients which are fortunate to have a Data Resource Management (DRM) or Data Administration (DA) department with the appropriate technology (case tools, data dictionary, metadata repository, procedures, naming standard's, politic clout with appropriate mission/charter statement) should be well positioned and may find these issues, less imposing. Those who do not address data quality will experience as I have, lack of data quality planning to be an evil that must be pursued. Failing to address the issues with conviction, or in the very least prepare by realizing and addressing the implications, predictably results in hiding the evils and then wondering why the data is incorrect in the DW.

1.1 Credits and Acknowledgments

This document was written from work performed on past projects, a variety of inputs and suggestions from within the data warehouse practices and various reference material.

I would like to thank **Ken Orr, Data Warehouse Institute** for stimulating my interest and instilling confidence for actively pursuing this paper and for writing the preface section.

Reference material/books used are acknowledged within the document as well as a consolidated listing in Appendix C.

2. The Politics and the Concerns

..... *The Politics*

Individuals/Clients pursuing a DW project who understand and are willing to take data quality seriously and take the appropriate steps "before" they have no choice, are receptive to pursuing data quality steps. The many who are not, see such items/considerations as being extra work of which the day has not come yet, do not want to rock the boat, or who want to pass the buck to others to clean up the mess at some later time. This occurs primarily due to knowing the data discrepancies must be researched, resolved, physically corrected (in actual file or through a extraction program), reprocessed and reloaded. In addition, once the data discrepancies are known, one must remove the cancerous data problems, which predominantly occurs up stream in the operational application systems. Usually these corrections to data quality occur outside of the domain, and or internal politics associated with the data warehousing project. Often the issues were not known, discussed, and therefore not seriously considered by the sponsor. Since the mechanisms required to resolve this problem vary in process, procedure, in time, effort, resources, consequences and of course cost, it is not a pleasant thought to consider - so out of site out of mind has a tendency to prevail.

..... *The concern(s)*

For those whose role is to sell and justify a DW to a organization (vendor or client), introducing such pointed and focused knowledge of cause, effect and resolution may scare the prospected sponsor and/or slow down the sales cycle and project before it has started, so this must be handled carefully. Most of those who sell the DW project (internal or external) who I have dealt with, mention data quality/validation with regards to knowing about it but move off that point as fast as possible. If the sponsor is cognizant of this and desires your assistance regarding having you address and rectify data quality issues (more for initial DW population verses ongoing population) associated with the DW as well as proceduralize the tasks for ongoing processing. It would be recommended to consider this opportunity on a time and materials basis or if it is a fixed price opportunity, only provided estimated completion dates until more is known (leave your self an opening). Getting into Data Quality analysis and resolution cannot be fully understood or scoped until you are actually into it, anything else is at best a guesstimate(guessing about a estimate). There are too many hidden gotcha's. It is like the old saying about the Greek myth of Pandora's Box. Everything is all right and everybody is happy until Pandora opens the box, letting all the evils out for everybody to see and thus deal with.

When data quality is classified in a more detailed manner the definitions are many. Each DW practitioner has his or her own horror story as well as the definition to this item. For me I define it less as a definition and more as its impact to that which I must do or have done within a project when I discuss this topic. I see dealing with data quality issues within the scope of an initial DW project as adding additional risk, additional tasks, additional time and resources, additional discrepancies and friction between the those who must do, those who must approve, and those who must accept.

..... *Pointing of the finger - Who is at fault*

One's ability to accurately extract/transform/validate the data that the DW consultant can obtain from the systems identified by a internal/external client will be up for scrutiny. It will also impact ones ability to accurately validate the data extracted and loaded into the data warehouse, thus impacting users willingness to accept the results. It has been my experiences that while the concept of junk in and junk out is well known, and on the surface accepted, the initial blaming finger will be pointed toward the person who packaged and created the DW. It will be their responsibility, in the eyes of the client, to identify and resolve data quality issues, until the DW consultant is able to identify the cause of the errors. This is a time consuming process with substantial analysis requirements starting from the point of contact and review (i.e. the DW); back through the database loading facilities, scripts, and environmental conditions; back through the extraction programs, to the sourced OLTP files; and potentially back into the operational applications, which originally created the data. Until this cycle occurs, a true understanding of the cause may not be fully known. As the DW consultant on the project, it is I who will have to burn the midnight oil, perform the fire fighting and face the initial wrath of from both sides until this is resolved and or proved it was not a result of our doing, the implementers of the DW project, it was the data that **I was told was 5% or less dirty.**

3. Project Scoping - Data Quality Considerations :

Data quality surfaces as one of the major stealth issue (out of sight, out of mind) above and beyond any others. It has a tendency to fall into a pseudo non-technical category which has a tendency to slip out of the domain of those involved in the DW project. Though it may be sensed that it is or will be an issue, unless properly scoped, we are not in the best position to put into place at the on-set of the project the necessary mechanism, organizational infrastructure, or application processes necessary to identify and rectify data quality problems that may impact the project.

.....*The process*

It is inherent in the process one follows in project scoping that there is a need to address, up front and to others, what the perceived risk and issues are which may impact the estimated length of time of a DW project and associated deliverables. This is always put to the test during the initial scoping of a DW project. Those who must sell the project (vendor or client) to others, seek to address were possible those issues that can be qualified and quantified so as to seek for themselves and to convey confidence to others that it is either achievable or that can be overcome with minimal challenges or obstacles. In doing so an appearance of control and confidence is conveyed to ensure others that the stated deliverables will be forth coming with in the estimated time. Unfortunately until the problems surface, and project leadership knows or feels data quality is a problem, addressing data quality is viewed as impacting or placing the project at risk. Accordingly, there is limited justification to it to take the necessary proactive process.

Those challenges or obstacles that do not fall under the above category are not dismissed, but are touched on lightly and were possible, quickly. This is necessary so as not give the appearance that the issues are not being dismissed or are unknown. In doing otherwise, what one might try and classify as an oversight, might tarnish ones image of being intelligent on issues which others might in the near future bring up. Until the problems surface and turn from they know or feel it is a problem, to it is impacting or putting at risk the DW viability, there is limited justification to it take the necessary proactive process. Appropriate scoping highlights such concerns earlier verses later when it may be too late.

.....*Things to consider*

Knowing the challenges one faces trying determine the risks, time and resources required, and to provide more appropriate estimates, requires engaging those who you must respond to, in a dialogue. This is preferably in the form of written caveats such as 1) a proposal, 2) a Statement of Work, 3) guidelines, 4) FYI memo, that indirectly says I told you so or I am educating you on the issues and risk of data quality and it is recommended you take notice and or act upon it.

.....*Things to do - (Questions to ask)*

Ask the champion, identified designate and or IT staff members, selected questions which attempt to gather information on what they understand about their applications, data sources, data entry facilities, process the data goes through, departments involved (the more people the higher the risk) and known data quality concerns/problems from a high level to a

low level based on your audience and available time. Where they cannot give you specifics on some questions, consider requesting them to quantify their response in the form of high/medium or low. I coin the result as a guesstimate. Though this is not as specific or quantifiable of response as one would like, it is all one normally can hope for. This though in itself can raise red flags as well as highlight potential issues.

In **APPENDIX A** I have listed a series of data quality **scoping questions** and data quality **characteristics** that can assist in giving one a feeling of the state of the data involved. **One does not expect** from these scoping questions full answers, nor all answers responded to, nor necessarily 100 % correct answers on the questions. Just more than one has to go on at the beginning. What you are told during a cursory meeting with the sponsor or client does not qualify as enough, though others may try to impress upon you it is.

Without any of these question being answered one has limited ability to reduce the risk and provided a closer guesstimate on the time and resources requirements as well as the complexity and potential hidden issues/gotcha's that we may encounter in the course of the DW project.

4. Data Quality Problems:

When talking with clients and those pursuing the implementation of DW projects, Data Quality when brought up is addressed more along the lines of it is a concept versus specifics (types of problems). The normal response after this when asked about how bad the quality of the data is, is normally stated as minimal or very low. On the other hand, when one qualifies the type of data quality problems that one can encounter, that perception increases drastically. It is not uncommon to have to reconsider the possibility of not populating or retaining a large percentage of data loaded into the DW due to concerns over the validity, cleanliness, accuracy of the data and at time the desired data may not being able to be found to be populated.

Data Quality problems should not only be considered for the current data extracted during the initial DW population effort, it needs to be considered for past data as well as future which can feed into the DW. The **current data** represent what was initially held in the OLTP systems prior to the initial population, which normally is limited in size. A banking environment normally holds 2-3 months of data with the rest backed ready to be recalled. **Past data** represents selected master and transaction files which normally mirror the current OLTP files structure that have been backed up ready for recall (with special programs/process in place). **Future data** represents the incremental data changes that have occurred (mostly to master files and cross reference tables) which must be captured after the initial population (normally a new set of extraction programs/process).

Listed in the below matrix is a comprehensive listing of core of data quality problems one may face through out the DW development process. Detailed descriptions of each error type are listed in **Appendix B**. If you know the possibility that certain errors exist, you will be more prone to spot them and to plan your project to attack the errors in a manageable way and with more focus :

Error Category	Types
Incomplete	<ul style="list-style-type: none"> - Missing records - Missing fields - Records or fields that, by design, are not being recorded
Incorrect	<ul style="list-style-type: none"> - Wrong (but sometimes right) codes - Wrong calculations, aggregations - Duplicate records
Incomprehensible	<ul style="list-style-type: none"> - Multiple fields within one field - Weird formatting to conserve disk space - Unknown codes - Spreadsheets and word processing files - Many-to-many relationships and hierarchical files that allow multiple parents
Inconsistent (most numerous)	<ul style="list-style-type: none"> - Inconsistent use of different codes - Inconsistent meaning of a code - Overlapping codes - Different codes with the same meaning - Inconsistent names and addresses - Inconsistent business rules - Inconsistent aggregating - Inconsistent grain of the most atomic information - Inconsistent timing - Inconsistent use of an attribute - Inconsistent date cut-offs - Inconsistent use of nulls, spaces, empty values, etc. - Lack of referential integrity - Out of synch fact (measures/numeric) data

Source : Larry Greenfield

When do Data Quality Problems occur :

Knowing what type of data quality problems can potentially exist can help point you in the direction of addressing the areas that may require the most attention. Knowing when the Data Quality problems occur allow you to take the knowledge of the DQ causes and become proactive in targeting the resolutions in advance to where they have a tendency to show up most and procedurelize if appropriate.

Due to these problems occurring at all stages of projects, in any part of a business process and for a variety of reasons, it is essential to identify them as a starting point of your focus. The below table provides a breakdown of "when, where and why" they occur. It is a useful, although not exhaustive, guide to most of the major causes of data quality problems.

When/Where	Why
System Conversions, Migrations or Reengineering	Inadequate data quality testing on conversion process. Conversion programs introduce new errors. Reengineering does not consider data context, usage or definitions
Heterogeneous System Integration	Data is inconsistent or contradictory across systems. Inadequate data quality testing on integration
Post-Integration of Heterogeneous Systems	Data remains inconsistent or contradictory across systems. Subtleties of poor data quality arise as new scenarios develop
Production Software	Software requirements were incomplete or errors were introduced in the development process. Lack of applied software engineering or production controls.
Database Design	Record and field definitions are too loose, unstructured or are not normalized. Schema lacks sufficient validation, and integrity rules.
Data Aging	The company cannot track the age of data, or has no program to update or enrich data.
Customer (UN-)Response	Data never fully captured. Customer form is badly designed, or no incentive is given to customer to offer response.
Fraud	Physical and logical system security is lax or compensating controls are absent.
Systems Internationalization	Overlapping or inconsistent interpretation or usage of codes, symbols, formats due to national differences.
Input Error	The system input method is badly designed, or lacks automatic validation. Human errors easily introduced.
Business Rules	System requirements lack adequate or current reference to business rules for data.
Policy and Planning	Lack of management attention to data quality management.

Source : *Chris Firth*

5. Data Cleansing Considerations - Manual Orientation

Data cleansing is the process of extracting data from the system of record source files, conditioning or reconditioning it to a level of acceptable quality based on pre-defined rules, and populating it into the warehouse. It includes analyzing data to discover its most appropriate meaning or use, standardizing the data into its lowest level of detail, identifying and consolidating duplicates, calculating derived and summary data, and finally loading the data into the warehouse.

This activity normally occurs during the initial data extraction. Later once the data has been populated into the DW, it should be performed on a regularly scheduled basis with selected integrity checking and data cleansing routines being run. This will report on inaccurate data (if your reporting mechanisms are in place) and perform various levels of data cleansing based on the process that have been created.

Main components of data cleansing are :

- **Data examination** determines the quality of the data, the patterns within it, and the cardinality of the fields (the number of different fields used).
- **Data parsing** determines the context and destination of each component of each field.
- **Data correction** matches the data against known lists (usually addresses, but sometimes master lists such as recipient databases) and ensures that all fields are tagged as good, bad, or automatically correctable. However, it is an understatement to say that this last category might involve many value judgments in the software design. Whenever possible, the conversion team should work with the customer to correct the data at the source.
- **Record matching** determines whether two records (perhaps of different types) represent data on the same object. This process involves many value judgments and requires sophisticated software tools.

Core places to target the cleanup of the data is :

Where	Implications
At the source system	Most difficult during scoping of project, but provides biggest impact ongoing effort and impact. Normally a forced issue and after the fact situation.
During extraction	Is mostly targeted during the DW project if requirements have been previously identified, which is normally limited due to time.
During Data Load	There are limited facilities and flexibility in the variety of activities that can be done here.
On the DW server after the data is loaded	Most visual location, though this is a after the fact clean up

Additional guidelines for data cleansing :

- Start small with an important, yet manageable, group of data. Not all data has the same value or quality issues. Focus first on the high-payoff data.
- Identify the system of record (authoritative source) from the legacy data sources by data groups. Data about a single object, such as customer "J. Jones," may exist in many files. Customer data related to ordering (i.e., a shipping address) may come from the sales file, while accounting data (i.e., a billing address) may come from an accounts receivable file. Customer profile information initially created in the sales file may have its authoritative source in the marketing file, where it is maintained.

- Identify the database for each data group that has the most vested stakeholder. Where data is maintained in multiple databases, identify the database where business processes are most likely to maintain current values.
- Analyze and discover the meaning, values, and business rules associated with the source data. This identifies apparent current uses and business rules.
- Inconsistency in the errors are the most difficult to address. It is advisable to conduct a baseline physical data audit to discover the accuracy of the data to get an idea what is and is not acceptable so you know what to expect. The physical audit compares data values with the real world those values represent. A random sample of 150 to 500 records from even the largest databases will provide a reliable assessment of accuracy. Compare data values with the sources: to verify customer data, contact the customer; to verify product dimensions, measure product samples.
- Automate as much as possible.
- Develop transformation rules carefully and test outputs.
- Involve knowledge workers and data producers in the physical audit and cleanup. This helps to generate a proactive defect-prevention culture.
- Clean data at its source database if the records are still used. Do not just clean data for propagation to the warehouse.
- Track time and costs involved in data cleanup. Data scrap and rework costs easily justify measures to prevent defects.
- Determine threshold values (Max/min) to check for
- Look for values that are not within specific codes/ranges/data types.
- The time, effort and resources spent trying to analyze and check for the data errors, potentially will require more time than it takes to rectify the original problem.

Source : Larry P. English

6. Error Handling and Reporting

This section identifies sample error handling and reporting facilities with examples and samples. This is where most of the ongoing creativity and cleanup time is spent.

Potential Data Quality Error Handling Levels

Describes codes that can be assigned to the type of error with the associated action taken after noted. In essence when a error is encountered, the record (or parts of it are written off to a error file) with the error being allowed or rejected. After that, a report is created with the error types reviewed with some action taken on it. Most of the time, the reporting of the errors enforce the business case to do something about the errors up stream verses pass the buck to the Data warehouse processes to clean up. Visualizing the problem with specifics always helps.

Error levels to consider :

Level D (assign default): If the error can be identified, correct and then populated into the DW. Allow the record to be processed but change the value of the data within the field(s) to a pre-defined default value. There is no need to record correction in an exception file. (i.e. handling of unexpected nulls, spaces, zeros, codes,..)

Level A (allow error) : If the error can be identified , but the data is **not important enough** to deal with regarding trying to correct the data before loading it into the DW. Load the data, make note of the exception and address it to the appropriate OLTP system head for resolution. (i.e. Limited, meaning less, or if any description was provided for a description field). Potential post load modification can be considered (i.e. SQL Update command)

Level F (fix error): If the error can be identified and the error is **important enough** that the data must be corrected, populated into the DW and record the error for future consideration. Write the record to a exception file, change the data value in the field(s) while retaining old value as well, load the records (minus the old value) in the exception file to the DW.

Level R (reject error) : This is a serious data error and “can not” be resolved within the extraction program and cannot be allowed to be loaded into the DW. Identify and document the error and address to the appropriate OLTP system head for resolution. Re-extract after resolution (i.e. Invalid Key).

Potential Error codes within Error Levels - For records processed that are in error :

Error files will be generated based on the source input records being read/used which have been assigned a error code. When these are cross referenced with a legend which describes the codes, these can be used to identify the type of error. This can also substantially reduce amount of detail that would other wise need to be included in the error file to describe the error.

Column	Error Code	Description Of Processing
xx-field-DESCRIPTION	D	<ul style="list-style-type: none"> • Meaning - No description was available for the input field. • Disposition - Default value is moved to Field and record is processed
xx-effect-DATE	F	<ul style="list-style-type: none"> • Meaning - Effective-date-YYMMDD date field is invalid • Disposition - Format of Source Date must be changed to match Target Date Format.
xx-cust-nbr	R	<ul style="list-style-type: none"> • Meaning - Customer number did not exist on the Customer table.
xx-div-ID	5	<ul style="list-style-type: none"> • Meaning - Division id was equal to zeros.
xx-item-nbr	6	<ul style="list-style-type: none"> • Meaning - Item number was equal to spaces or low values
xx-vendor-nbr	7	<ul style="list-style-type: none"> • Meaning - Vendor number was equal to spaces, low values or zeros (same as error code 6 but different field)
xx-group-cd	8	<ul style="list-style-type: none"> • Meaning - Group code was equal to spaces or low values. • Disposition - Record is processed and NOT written to any error file. Existing Logic is currently commented out in the xxx process which will reject this type of record (per xx/xx/xx email from specific person) .
xx-class-cd	9	<ul style="list-style-type: none"> • Meaning - For specific source file (xxx.xxx) only - Class code was equal to spaces or low values.
xx-expire-date	10	<ul style="list-style-type: none"> • Meaning - Expiration date was not valid • Disposition - Record is processed with a default date of 9999-12-31 and is reported on the error file
xx-xx-xx	13	<ul style="list-style-type: none"> • Meaning - Additional types of error that can be encountered. Add to this sample list. • disposition - Record is ?

Error and Data Population Reporting Considerations

Listed below are reporting examples that if created can assist in identifying errors that have occurred during the data transformation process as well as understanding the status of what has and had not been populated into the D.W tables. Additional overall summary statistic reports should be considered to depict the implications of the errors.

..... Rejected rows in the DW Error tables

1. Generate a statistics reports of the **Rejected rows in the DW Error tables** after each Transformation \ population cycle has occurred. This would include :

A. Row Count of Errors and description within a Severity level for a given DW load/population attempt listing accumulated errors (i.e. number of Columns in error) per severity level and Code for a given DW Error Table, Grouped by Application Id. This will be used by the D.W Staff.

Appl. Id	Error Table.	Severity	Error Code	Nbr. of Rows	Error Desc.
GRD	XXXX	Reject	1	10000	X-ref Field not found
GRD	YYYY	Reject	2	5000	Missing or Invalid data
CLS	ZZZZ	Reject	1	100	X-ref Field not found

B. Same as (1A) but individuals reports would be generated and provided to each Operational Owner, listing statistics per error table per application which belongs to the operational owner. This would be accompanied by a detail listing of the records in error as well (see #3).

.....**Completeness and Validity of populated data**

2. Generate a statistics reports of the **Completeness and Validity** of the data that has been populated into the DW for a population\load cycle. This would attempt to identify :

A. Number of rows populated verses those that were rejected. This could show as accumulated totals as well as a quick percentage (per table). This would be used as a indication of what percentage of the DW table was not loaded during a particular population cycle.

DW Table	Nbr. of Rows rejected	Nbr. of rows populated	Percentage Populated
Loans	15000	750000	98%
GL	520	10000	94.8%
DDA	25000	65000	61.5%

B. A listing of the columns in error that were not load (as a percentage) as compared to those that were, within a DW table. This would highlight which columns have the highest error rate with the records.

C. A identification of the Referential Integrity Violations that have occurred. This would be related to those key columns that were in error due to not finding their related parent or cross reference file rows. This would occur for these records that were rejected due to not satisfying the cross reference validation check (i.e. table lookup). Though this might also be recorded in item 2B, should additional validation checks be introduced, the segregation of this statistics as a separate reported on error would provide more visibility to an important statistic.

Note : When or if further validation considerations are pursued within the transformation process. The information about the columns in error could be elaborated on to provided a more detailed break down of what is invalid (i.e. missing, invalid, spaces, nulls,..), what was allowed, what was accepted with defaults. This could then be grouped and presented as a Data Quality report per field.

.....Detailed Reports of Rows in Errors Table

3. Generate Detailed Reports of rows contained in the Error Table.

- A. A listing of each rows per error table would be produced. This report could be used by both the DW team as well as provided to the Operational Owner on a as need basis.

Note : If the number of rows in to be reported on are to large, the initial generation of this report may need to be re-considered and a more focused approach pursued. This may lead to consideration for add-hoc access for query purposes against the Error table in question with report generation based on that which has been queried.

7. Data Quality Improvement & Resolution

As discussed at the beginning of this document, Data quality once exposed during the Data Warehouse project, is initially considered a issue for the Data Warehouse project to resolve. This is primarily due to end-user queries highlighting discrepancies that were not or could not be exposed when the data was initially extracted or that were hidden in the OLTP system. Due to this, the perception is to rectify the errors at the point of notice or query within the data warehouse. Unfortunately repairing the error in this fashion only represent a temporary solution with the effort and time requirements to continually deal with the data verification and reconciliation requirements outweighing the benefits. There are exceptions, but this is normally the rule. Even after the problem is rectified within a DW processing point or within the DW data itself, the original OLTP system will continue to send the same or variations of bad data. Due to this, the errors must be resolved at the source, the application system or processes involved prior to feeding into the system of record where you have extracted the data from.

If data reconciliation and resolution is not addressed during or after the initial DW project other than as patches to a evolving DW process, the errors start compounding themselves with the user becoming more disturbed with the achieved results. Only when it gets bad enough will the necessary focus and effort be considered and applied. This normally implies resolving and or reconciling the problem upstream in the OLTP system verses within the DW process. Core areas of consideration to be targeted initially are those processes associated with data-entry, those process/programs that are fed the results of this process or those data that goes through a series of processes and departments before residing in the final system of record which you extract from. Additional consideration should be given to those application systems which have a high degree of on-line or batch errors and reconciliation requirements showing up as well. Though the data may enter into the processes and may not have come in as an error, they can become a error as a result of unexpected OLTP processing issue such as scheduling conflicts and recovery requirements.

If this is not resolved at the source :

- Your DW process will continue to receive and fix the same data error over and over again
- The data being feed back into the OLTP system (i.e. Operational Data Store) could cause replication in other areas unknown to you.
- If the data is considered relevant for analysis against the OLTP system or as an extract. (i.e. aggregated /derived values used) you will not know which source is correct, if differences show up when compared.

Examples of how varied the reconciliation and improvement activities may be :

- One the data is in the DW, review and change the data as a add-hoc fix within the DW as a interim solution to insure that the initial population of the DW will finish on time,. This will also show initial value to those concerned. Though normally, the DW practitioners are scurrying in the back ground trying to resolve the data quality issues before the incremental populations are ready to proceed.
- To attempt to address the cause of the problems one may choose not to resolve in the DW, but to reject the discrepancies and report on them. This provides the visibility as well as visual proof of the errors to add credence to the need to resolve at source system which is normally outside of domain of the initial project. After all, if it's bad in the warehouse, it's probably bad all the way back in the source system from which it came.

Areas of responsibilities in solving data-related problems :

In pursing Data reconciliation, various areas and skill sets are required. The below matrix outlines a basic framework to start with at a clients site.

Type of data problem	Area of Responsibility						
	Tech Support	DBA	Appl Devl	System User	Data Admin.	System Controller	Quality Assurance
Technical	X	X		X			
Software	X	X	X	X			
Operational	X		X	X		X	
Integrity/missing data			X	X	X	X	X
Inaccuracy/Unreliability			X		X	X	X
Built in			X	X	X	X	
Data Definitions/Format Mismatches		X	X		X		
Inaccessibility	X	X	X	X		X	
Procedural					X	X	X
Incompatibility			X		X	X	
Authorship			X		X	X	

Source : Brian Horrocks and Judy Moss

Pursuing the improvement of Data Quality :

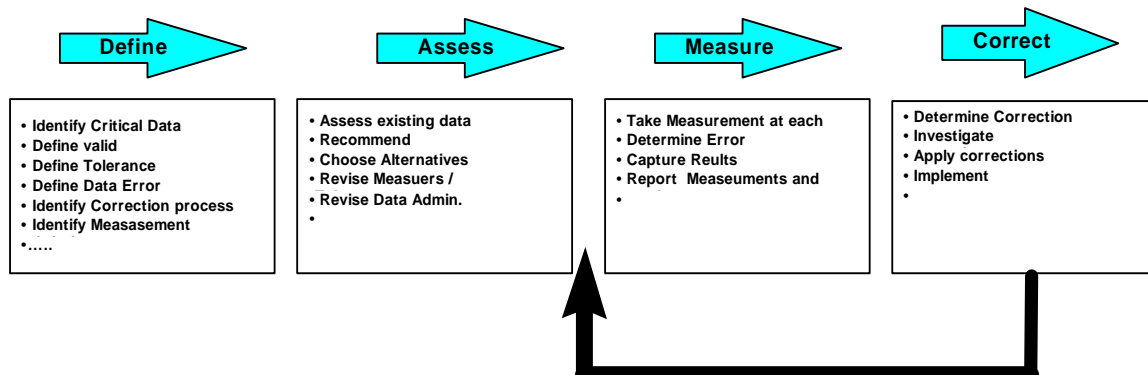
1. Determine the initial / critical business functions to be considered as is associated with the population of the DW
2. Identify criteria for selecting critical data elements, which if the data is not accurate, timely, complete, or consistent will negatively impact the business. Examples of this are :
 - Halt Business processing or cause unacceptable delays
 - Result in a substantial misapplication of resources
 - Create a significant legal risk to the business
3. Designate the critical data elements
4. Identify known data quality concerns for the critical data elements, and their causes. Ask the below questions and consider their characteristics (see Appendix A).
5. Determine the Data Quality standards to be applied to each critical data element, which includes defining the user's expectation for the data (i.e. accuracy, legibility, completeness, consistency, timeliness,...).
6. Design a measurement method for each standard.
7. Identify and implement quick-hit data quality improvement initiatives
8. Implement measurement methods to obtain a Data Quality base line
9. Assess measurements, data quality concerns, and their causes. Examples of the causes are :
 - clarity and understanding of data definitions
 - business forms, procedures and workflow
 - employee training on job responsibilities and tasks that impact data quality
 - automated processing capacity limitations and delay
 - mis-designed software
 - error correction procedures
10. Plan and implement additional improvement initiatives
11. Continue to measure quality levels and tune initiatives
12. Expand process to include additional data elements
13. Link data quality improvement to specific business objectives
14. Utilize Senior management to drive improvement initiatives

15. Balance Long term vision with achievable short term objectives
16. Tackle the easiest problems offering the highest payoffs first
17. Coordinate data quality efforts across business functions using the same data
18. Do not create blanket quality standards
19. Create plans that are achievable within the likely available resources
20. Actively involve business staff at multiple levels
21. Use proven methods and tools
22. Bring in and leverage outside expertise
23. Expect traditional business polices and practices to be a major contributing factor to data quality problems
24. Provide early orientation and training to those involved in a data quality improvement initiative
25. Establish a safe environment for he discovery of problems and their causes
26. Establish a core team or support group
27. Establish qualitative and quantitative reporting of program results
28. Provide incentives and recognition that go beyond the usual

Source : Dennis Berg / Christopher Heagele

Data Warehouse Data Audit/Control Methodology

- > > **Define** the process/data that requires
- >> **Assess** the existing process/data
- >> **Measure** and record/report each acquisition run
- >> **Correct** failures and re-process



Source - Lowell Fryman

8. Mechanical based validation during Data Migratio

Aside from approaching validation within the Data extraction and transformation activities one needs to consider validating the results as they are migrated and populated into the data warehouse. This is were the data which has been transformed and massage is being positioned to be reviewed.

I have attempt to identify mechanical based items that could be useful for validation checking during a sample processing cycle of the data warehouse. This is normally the least costly in time and resources to put together and execute on a routine basis.

The important and time consuming user based validation checking is not discussed here in detail. This normally involves the client putting together business based validation of rules computations / formulas / aggregations /summarized values,..) and relationships (i.e. how many accounts does a particular customer have, are his debits/credits correct - R.I based issues) and comparing them against a OLTP based set of reports or validation data they have selected. Normally this involves a reasonable amount of lead time for the user to put together and validate. Though this is normally told on the onset of the project to the user before it is needed, it is rare that this comes together comfortably, it tends to be a last minute thing which unfortunately can impact the acceptance test procedures, so plan for it.

Data Validation Reporting Information considerations

Source Staging process - Extraction file (from source platform) to => Teradata Staging tables.

- **Count of records** from Source file is performed and stored/recorded. May consider a trailer record (at EOF write a final record to contain counts) on Trailer record in Staging file.
- **Count of rows** in Staging Table (after load) is performed and compared against input Staging Files record count.
- **Accumulate totals** are calculated for each field (Source Staging file and Target staging table) that contains a currency value such as Amounts, Balance, Limits, Totals,... There is no consideration for type of currency associated with the currency value. These values are combined and compared per field against the Input Staging Files fields.
- **Identification of Duplicates** would occur after Data is loaded into Staging tables

Note : Additional information (count/values) would have to be added to the Source files trailer record, such as *accumulated totals*.

Transformation process (after initial load) = >Teradata Staging tables to Teradata target and error tables.

- **Count of rows** is performed against the Target and error Table(s). These values are combined and compared against the row count value in the Staging Table.
- **Accumulate totals** are calculated against the Target and error Table(s) for each field that contains a currency value such as Amounts, Balance, Limits, Totals,... These values are combined and compared per field (i.e. Target fld + Error Fld = Total) against the Accumulated amount contained in the Staging Table. There is no consideration for type of currency associated with the currency value.
- **Identification of Duplicates** would occur after Data is loaded into Staging tables (optional ?)

Miscellaneous (Generic Considerations)

- Count input rows, input rows dropped due to errors, not processed, and resulting output rows
- Compare records input and records output for process step
- Validate Total Dollars (Values) Input, dropped, output, Sum total
- Count the number of occurrences (i.e. Occurs clauses/fields or potential different values in field for an attribute)
- Validate Referential Integrity - Primary / Foreign key match (lookups /Load)
- Identify tolerance levels, that if exceeded warrant review
- Compare selected values/totals of this load period against the last processing period
- Duplicates - Consider during/After initial Loads
- Consider looking at records for specified condition (i.e. Gender = M,F,0, 1,..)
- Look at only previously selected/qualified records for specific conditions
- Look for data where field is alphabetic vs. Numeric, but not both
- Create threshold values to watch for (i.e. counts, values, ..)
- Consider completeness checks : If you can continuously identify a set number of items to measure (i.e. control totals) at a ending period or point in time of a activity, count the number of items and compare with the like totals from yesterday to see how close/complete the values are.
- Consider a Reasonable check : determine a appropriate range that a selected value or aggregated value should fall into and perform a check against incoming data values to determine if it is within a range. Also compare it to past values of previous extraction's.

9. APPENDIX A - Data Quality Scoping Questions

- What level of customer data will be extracted first (i.e. Enterprise Banking top customers (few), Wholesale, Retail (most customers),...).
- Location of where data physically resides (i.e. different cities/bldg.'s/platforms/PC's). The more locations the higher potential for more Data quality problems.
- Any external data used (I.e. outside data source providers - Reuters,...). May cause lack of control or prior review of the quality. Must take it as it is with no opportunity to clean-up.
- Is there a consistent CIF key field among the banks customers which allows the banks different departments the ability find the customer and related data, or is the customer referenced by several different keys/fields by different banking applications.
- Does each of the Business Unit - BU (Enterprise, corporate, Wholesale, Retail, subsidiary) have a CIF?. The more pseudo non-centralized CIF's the higher the potential for disparities to occur.
- Before a customer is created, do they create a CIF no for a customer or is it automatically generated by a application system without human interference, thus limiting the potential for duplicates or other undesirable anomalies.
- For each application/product, what is used as the customer key and how is it linked to the CIF system (application CIF or central CIF) such as *Customer number, account number*. **Different keys could be generated/used by different systems.**
- Does the client have a Data Admin. Department. This may indicate a level of formality and a potential centralize point of reference and control point.
- If there is no DA dept., are forms of such activities performed by the DBA group (if no DA dept.) or by the application groups or a mix.
- Identify if they have a data dictionary (not RDBMS system catalog). This may help indicate the quality of standard meanings and potential data quality controls.
- Is the bank having to deal with Y2000 date conversion that may effect the data held on a customer. Because there application systems are only 8 years old, this may not be the case, but ?.. There normally are many date based fields involved which add to the complexity and time requirements if we have to deal with it. I would hate to be doing there year 2000 conversions for them and not factoring it in.
- Level of Data Availability in tables/files or source systems (**high**, med, low) based on what they want to see. Assumes they know where for example profitability based data comes from - app. packages/excel ***spread sheets***.
- Are there key pieces of customer data **missing** in the OLTP data which is required to be added to the source data for current reporting practices (H/M/L)

- Amount of extra data necessary to complement/enrich (derived, aggregated, missing non-OLTP data) the data source for current reporting practices, such as through data entry mechanisms -(H/M/L)
- What is the quality or confidence in data correctness (good data vs. Bad) in source (i.e. date related, standard codes to match on, ...) - (H/M/L)
- Any known data scrubbing/cleaning activities required on source systems data before the data is saved (H/M/L)
- Any issues with handling different data types (I.e. Double byte chars,..) (H/M/L)
- Are there customer & address matching issues/errors (H/M/L)
- Is there much reconciliation requirements to get the financial figures (depends on what business matrix are being reported on) correct on either a monthly or quarterly basis (H/M/L)
- **What type of reconciliation keeps occurring.** Can they give the normal reasons/cause for this. Though it is their responsibility to clean up the data, they may not know if they caused it or we caused it in the DW easily. So if we know in advance we can point it out and/or have them be proactive versus reactive when the totals do not added up. If not the finger will be pointed at us first to figure out.
- Can they identify how many sources / process (i.e. applications, teller, data entry process, check clearing, scanner,..) the data in a application may pass through prior to being considered the perceived system of record or final point of reference. (H/M/L) - The more processes, the higher data quality problems.
- How many departments use non-automated procedures to capture data (i.e. hard copy reports/forms) are used as the source data into PC based systems ?..)
- Who implemented the application (banks in-house resources/in-house vendor created/bought vendor app. package). Vendor packages have a tendency to leave in fields that are not used for the current client, have varied attribute domain types that are inconsistent with their use, lack detailed documentation to describe their file/files structures.
- How old are the Application. packages.
- Who maintains (develops/fixes/changes) the application system, is it a **IT** combined dept., individual App. Dept., Out Sourced to a vendor/third party,..)

In addition, one can complement Data Quality categories with further investigating Data Quality Characteristics as identified in the below matrix :

Type	Description
Accuracy	% of values that are correct when compared to characteristics of actual object described by the data
Addressability	% of data that can be understood verses parsing a word in to bits and pieces to understand (i.e. long name)
Completeness	% of data having values entered into it
Consistency	% of matching value conditions or derived value conditions satisfied.
Reliability	% of referential integrity conditions satisfied
Timeliness	% of data available within a specific threshold time frame (i.e. days or hours)
Uniqueness	% of records with uniqueness violations (duplicate primary key values)
Validity	% of data having values that fall within their respective range of allowable values

Source : Duane Hufford

10. APPENDIX B - Data Quality Problem Errors

Incomplete errors

- **Missing records** - This means a record that should be in a source system is not there. Usually this is caused by a programmer who diddled with a file and did not clean up completely. (I read a white paper about how users have to "fess up" about bad data. Actually, usually system personnel cause MUCH more headaches than users.) Note you may not spot this type of error unless you have another system or old reports to tie to.
- **Missing fields** - These are fields that should be there but are not. There is often a mistaken belief that a source system requires entry of a field.
- **Records or fields that, by design, are not being recorded** - That is, by intelligent or careless design, data you want to store in the data warehouse are not being recorded anywhere. I further divide this situation into three categories. First, there may be dimension table attributes you will want to record but which are not in any system feeding the data warehouse. For example, the marketing user may have a personal classification scheme for products indicating the degree to which items are being promoted. Second, if you are feeding the same type of data in from multiple systems you may find that one of the source systems does not record a field your user wants to store in the data warehouse. Third, there may be "transactions" you need to store in the data warehouse that are not recorded in an explicit manner. For example, updating the source system may not necessarily cause the recording of a transaction. Or, sometimes adjustments to source system data are made downstream from the source system. Off-invoice adjustments made in general ledger systems are a big offender. In this case you may find that the grain of the information to be stored in the warehouse may be lost in the downstream system.

Incorrect errors - data that is actually incorrect

- **Wrong (but sometimes right) codes** - This usually occurs when an old transaction processing system is assigning a code that the transaction processing system users do not care about. Now if the code is not valid, you are going to catch it. The "gotcha" comes when the code is wrong but it is still a valid code. For example, you may have to extract data from an ancient repair parts ordering system that was programmed in 1968 to assign a product code of 100 to all transactions. Now, however, product code 100 stands for something other than repair parts.
- **Wrong calculations, aggregations** - This situation refers to when you decide to or have to load data that have already been calculated or aggregated outside the data warehouse environment. You will have to make a judgment call on whether to check the data. You may find it necessary to bring data into the warehouse environment solely to allow you to check the calculation.

- **Duplicate records** - There usually are two situations to be dealt with. First, there are duplicate records within one system whose data are feeding the warehouse. Second, there is information that is duplicated in multiple systems that feed in the same type of information. For example, maybe you are feeding in data from an order entry system for products and an order entry system for services. Unbeknownst to you, your branch in West Wauwatosa is booking services in both the product and service order entry systems. (The possibility of situation like this may sound crazy until you encounter the quirks in real world systems.) In both cases, note that you may miss the duplicates if you feed already aggregated data into the warehouse.

Incomprehensibility errors - Types of conditions that make source data difficult to read.

- **Multiple fields within one field** - This is the situation where a source system has one field which contains information that the data warehouse will carry in multiple fields. By far the most common occurrence of this problem is when a whole name, e.g., "Joe E. Brown", is kept in one field in the source system and it is necessary to parse this into three fields in the warehouse.
- **Weird formatting to conserve disk space** - This occurs when the programmer of the source system resorted to some out of the ordinary scheme to save disk space. In addition to singular fields being formatted strangely, the programmer may also have instituted a record layout that varies.
- **Unknown codes** - Many times you can figure out what 99% of what codes mean. However, you usually find that there will be a handful of records with unknown codes and usually these records contain huge or minuscule dollar amounts and are several years old.
- **Spreadsheets and word processing files** - Often in order to perform the initial load of a data warehouse it is necessary to extract critical data being held in spreadsheet files and/or "merge list" files. However, often anything goes in these files. They may contain a semblance of a structure with data that are half validated.
- **Additional unexpected information** (potential stray characters - %, ^, *, _,...) in Text based fields such as in a Address.
- **Different levels of punctuation** in character and or numeric fields as (, -, !, :)

Many-to-many relationships and hierarchical files that allow multiple parents - Watch out for this architecture in source systems. It is easy to incorrectly transfer data organized in such manner.

Inconsistency errors

The category of inconsistency errors encompasses the widest range of problems. Obviously similar data from different systems can easily be inconsistent. However, data within one system can be inconsistent across locations, reporting units, and time.

- **Inconsistent use of different codes, flags, numbers or field (mostly textual) sizes** - Much of the data warehousing literature gives the example of one system that uses "M" and "F" and another system that uses "1" or "2" to distinguish gender. May I suggest that you wish that this is the toughest data cleaning problem you will face. These values may also change over time.
- **Inconsistent meaning of a code** - This is usually an issue when the definition of an organizational entity changes over time. For example, say in 1995 you have customers A, B, C, and D. In 1996, customer A buys customer B. In 1997, customer A buys customer C. In 1998, Customer A sells of part of what was A and C to customer D. When you build your warehouse in 1999, based on the type of business analysis you perform, you may face the dilemma of how to identify the sales to customers A, B, C, and D in previous years.
- **Overlapping codes** - This is a situation where one source system records, say, all its sales to Customer A with three customer numbers and another source system records its sales to customer A with two different customer numbers. Now, the obvious solution is to use one customer number here. The problem is that there is usually some good business reason why there are five customer numbers.
- **Different codes with the same meaning** - For example, some records may indicate a color of violet and some may indicate a color of purple. The data warehouse users may want to see these as one color. More annoyingly, sometimes spaces and other extraneous information have been inconsistently embedded in codes.
- **Inconsistent names and addresses** - Strictly speaking this is a case of different codes with the same meaning. My unscientific impression of this type of problem is that decent knowledge of string searching will allow you to relatively easily make name and address information 80% consistent. Going for 90% consistency requires a huge jump in the level of effort, Going for 95% consistency requires another incremental huge jump in effort. As for 100% consistency in a database of substantial size, you may want to decide if sending a person to Mars is easier.
- **Inconsistent business rules** - This, for the most part, is a fancy way of saying that calculated numbers are calculated differently. Normally, you will probably avoid loading calculated numbers into the warehouse but there sometimes is the situation where this must be done. As noted before, you may have to feed data into the warehouse solely to check calculations.
- **Inconsistent aggregating** - Strictly speaking this is a case of inconsistent business rules. In a nutshell, this refers to when you need to compare multiple sets of aggregated data and the data are aggregated differently in the source systems. I

believe the most common instance of this type of problem is where data are aggregated by customer.

- **Inconsistent grain of the most atomic information** - Certain times you need to compare multiple sets of information that are not available at the same grain. For example, customer and product profitability systems compare sales and expenses by product and customer. Often sales are recorded by product and customer but expenses are recorded by account and profit center. The problem occurs when there is not necessarily a relation between the customer or product grain of the sales data and the account - profit center grain of the expense data.
- **Inconsistent timing** - Strictly speaking this is a case of inconsistent grain of the most atomic information. This problem especially comes into play when you buy data. For example, if you work for a pickle company you might want to analyze purchased scanner data for grocery store sales of gherkins. Perhaps you purchase weekly numbers. When someone comes up with the idea to produce a monthly report that incorporates monthly expense data from internal systems, you'll find that you are, well, in a pickle.
- **Inconsistent use of an attribute** - For example, an order entry system may have a field labeled shipping instructions. You may find that this field contains the name of the customer purchasing agent, the e-mail address of the customer, etc. A more difficult situation is when different business policies are used to populate a field. For example, perhaps you have a fact table with ledger account numbers. You may find that entity A uses account '1000' for administrative expenses while entity B uses '1500' for administrative expenses. (This problem gets more interesting if entity A uses '1500' and entity B uses '1000' for something other than administrative expenses.)
- **Inconsistent date cut-offs** - Strictly speaking this is a case of inconsistent use of an attribute. This is when you are merging data from two systems that follow different policies as to dating transactions. As you can imagine, the issue comes up most with dating sales and sales returns.
- **Inconsistent use of nulls, spaces, empty values, etc.** - Now this is not the hardest problem to correct in a warehouse. It is easy, though, to forget about this until it is discovered at the worst possible time.
- **Lack of referential integrity** - It is surprising about how many source systems have been built without this basic check.
- **Out of synch fact (measures/numeric) data** - Certain summary information may be derived independently from data in different fact tables. For example, a total sales number may be derived from adding up either transactions in a ledger debit/credit fact table or transactions in a sales invoice fact table. Obviously there may be differences because one table is updated later than another table. Often, however, the differences are symptoms of deeper problems.

11. APPENDIX C - References

- When do Data Quality Problems Occur - Internet 1997 - Chris Firth
- A Taxonomy of data warehouse data errors- Internet 1997, Larry Greenfield - LGI Systems Incorporated
- Help For Data Quality Oct. 7, 1996 INTERNET - Larry P. English Information Impact International Inc
- Data Quality Part II - Internet , 1997 Duane Hufford (Principle Consultant - AMS)
- PRACTICAL Data Admin - Prentice Hall, 1993 - Brian Horrocks and Judy Moss -
- Building, Using an Managing the Data Warehouse, Prentice Hall -1997 Dennis Berg/Christopher Heagele
- Implementing DW Audit & Control Process - Oct. 1997, Data Quality Conference Proceedings - Lowell Fryman (Intelligent Solutions)