A Tutorial on Checking Data in a Database

DatabaseAnswers.org

10/6/2010

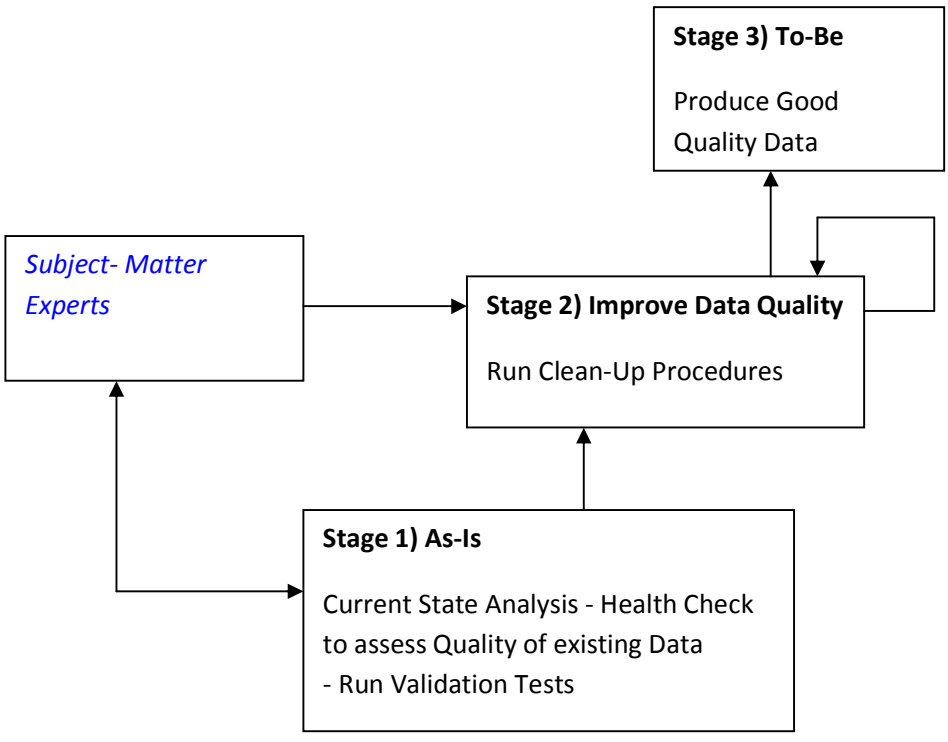
Barry Williams

A Tutorial on checking the Quality of Data in a Database

1. Introduction

1.1 WHAT ? – an Overview of the Process

Measuring and improving Data Quality is a three-stage process :-



1.2 Three Stages

There are three Stages in the Process:-

Stage 1 – Establish the 'As-Is'

- Carry out a 'Health Check' to assess the quality of the existing data – the 'As-Is'.

Stage 2 – Achieve the 'To-Be'

- Improve the Health until it reaches the required level – the 'To-Be'.

Stage 3 – Maintain good Data

- Put procedures in place to ensure good quality 'Healthy' data is maintained.

We start with a Current State Analysis to audit and provide a 'Health Check' assessment of the current situation, which establishes the 'As-Is'.

Running Data Quality Audit Tests is a repetitive process.

Running Data Quality Improvement Tests is even more repetitive because it involves defining Clean-Up procedure, (eg in SQL) and then running tests repeatedly and adding Clean-Up code between each Run until an acceptable level of good quality data is reached.

1.3 Legacy Systems

This Paper assumes that the Data we are interested in is all in Relational Databases that support SQL.

For Legacy Systems where this might not be true, the recommended approach is to explore the use existing Data Extract and Report facilities that could be used to generate CSV files for the required Tables and Fields.

These CSV Files can then be loaded into a Relational Database in a Staging Area where the power of modern technology can be employed.

2. Getting Started

2.1 Identify the Systems, the Data, the Stakeholders and the Goals.

We establish the 'As-Is' with a Current State Analysis to assess the current situation.

Then we talk to the Stakeholders to determine the target in terms of specific improvements in Data Quality.

This will take into account what is acceptable to the Stakeholders, because 100% might be difficult and expensive to achieve and not be necessary.

2.2 Engage with the Stakeholders

We will engage with all interested parties to determine the goals of the engagement, such as desirable improvements in any Data problems.

2.3 Approach to the Analysis of Data Quality - Health Check

2.3.1 HOW ? - Levels of Tests

Data Quality consists of running Data Profiling software to test data at five levels :-

Level 1. Individual fields which could be invalid.

For example, ID greater than zero, Names not blank and Start Dates not in the future.

If Postcodes are present in a Table, they can be validated against the Post Office PAF File, otherwise it is difficult to validate Addresses, especially if they are from outside the UK. It is necessary to identify the fields that can be tested and that it makes sense to do so. For example, a name should not contain any numbers but it might not make economic sense to check them all.

Level 2. Relationships between fields in a record.

For example, an End Date earlier than a Start Date

Level 3. Relationships between Parents and Children – 'Referential Integrity'.

For example, a Requisition from an invalid Unit or for a non-existent Product.

Level 4. Business Rules for specific conditions.

For example, a Tornado Aircraft cannot be ordered using a Requisition.

Level 5. General complex conditions.

For example, the total value of Requisitions for a specific Organizational Unit must not exceed a predefined limit. This kind of Test is out of Scope for a Health Check and might be difficult to justify in a more comprehensive check of a Database.

The critical aspect would be the existence of any Design Patterns that could be used to do large-scale testing for a reasonable budget.

Here is a (fictitious) example for a Customer record :-

CUSTOMER RECORD	Record ID (>0)	Name (not blank)	Date Registered (in the past)	Date of Latest Contact (Blank or >Date Registered)

2.3.2 Data Profiles

We can establish a data profile using standard SQL to check Date fields and cross-references :-

- Discuss the value of DQ Metrics with the Stakeholders, for example, % Orphan Requisition records
- Define the Metrics to be calculated and referred to as Key Quality Indicators or 'KQIs'.
- Evaluate the benefits of a Dashboard approach.
- Develop the SQL to measure Metrics, check Date fields and cross-references
- Run Tests to establish data profiles at five Levels described above.

1. Produce Health Check Report
2. This is a sample Quality Test Run Results Report to include Fields, Test and Record Counts. A more comprehensive Form is shown in Appendix A.

No	Table	Field	Data Type	Validation Test	Valid Record Count	Invalid Record Count (%)
A.1	Requisition	Date of Requisition	Date	>1-Jan-1990	10,000	0 (100%)
A.2	Requisition	Date Requisition Met	Date	=>Date of Requisition	10,000	100 (1%)

2.4 Improvement of Data Quality

1. Start with Health Check Report
2. Identify errors to be corrected in terms of improvements to KQIs and Metrics.
3. Determine clean-up processes in discussion with the Stakeholders and SMEs.
4. Run clean-up processes for relationships between fields in a record.

   For example, a blank field could be set to a default.

5. Run clean-up processes for relationships between fields in a record.

   For example, a blank End Date could be set to a default time after the Start Date

3. Run clean-up processes for orphans and other logical problems in relationships between different tables.

   For example, a Requisition from an invalid Organizational Unit or for a non-existent Product.

7. Determine enhanced clean-up processes in discussion with the Stakeholders and SMEs.

8. Repeat Steps 4 to 7 until the Data reaches an acceptable Quality.

2.5 Conclusion

Then the User signs-off the results and plans are made to ensure that good quality Data is maintained in the future.

3 A Case Study

3.1 The 'Things of Interest'

This diagram provides a simple overview of the principal functions in Logistics.

It is intended to be used a simple example that will be understood by most people.

It should be created in Microsoft Word, and it should be produced by the person checking the quality of a Data Model.

The purpose is to confirm the understanding of the Scope and functionality of the Data Model being

reviewed.

It should be created in not more than an hour to show the understanding that the person has acquired.

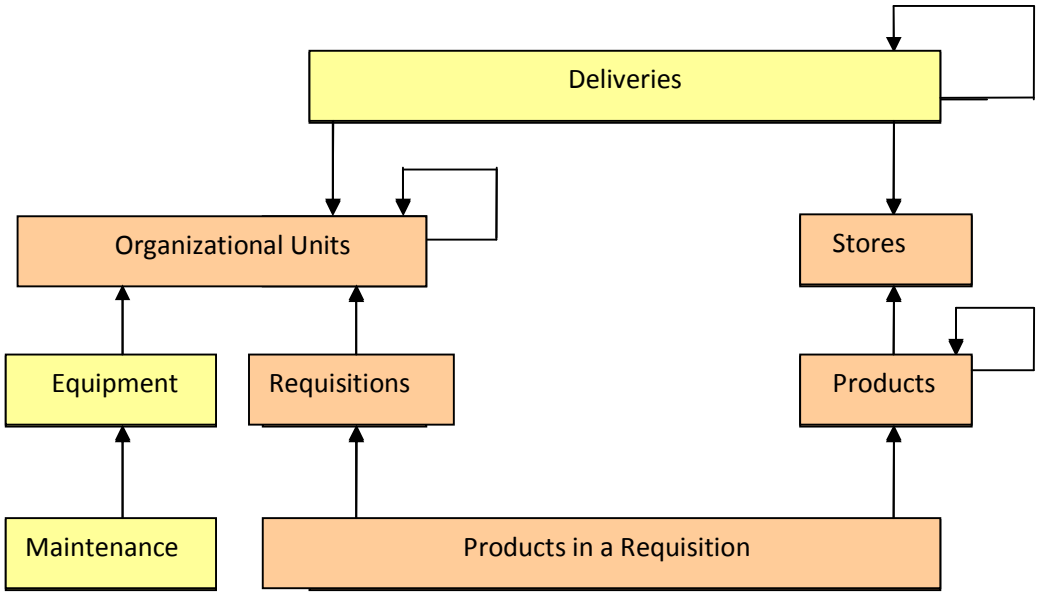
In other words, it is produced when the time is right, and not right at the beginning of the work.

This shows that Organizational Units raise Requisitions for Products to be supplied from Stores.

These are then Delivered to Organizational Units that are responsible for subsequent Maintenance of the Equipment.

The Maintenance Table would store details of Schedules, Maintenance work carried out, perhaps Spares involved and so on.

A Diagram of the 'Things of Interest' :-



3.2 Draft the Business Rules

Business Rules are valuable because they define in plain English with business terminology the underlying relationships between the Terms that appear in a Data Model.

The Stakeholders will then be able to agree and signoff the Rules.

Here is a small example.

No	TABLE	DESCRIPTION
D.1	Requisitions and Organizational Units	A Requisition must be raised by a valid Organizational Unit.
D.2	Requisitions and Products	A Requisition must refer to valid Products.
P.1	Products and Stores	Products are kept in Stores

3.3 Draft a Glossary of Terms

It is very important to establish agreed definitions of terms and words in common use.

This is a small example.

TERM	DESCRIPTION	COMMENT
Requisition	A Request or Requisition for Products or Materiel to be supplied to the Requesting Organizational Unit.	
Product	An Asset that can be separately ordered. It can be a Component and a part of a larger Assembly. It can be very small, such as a Washer, or very large, such as a Tornado aircraft.	Very large Products are normally referred to as Equipment.

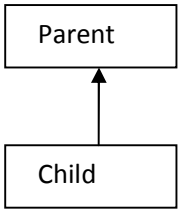
3.4 Sample Tests for Parent-Child Tables

3.4.1 The Simple Parent-Child Relationship

Parent-Child tables represent a common Design Pattern, for example, Organizational Units and Requisitions.

The general approach is to test for the underlying logic in Parent-Child relationships.

This simple Diagram in Word shows the two Tables involved :-

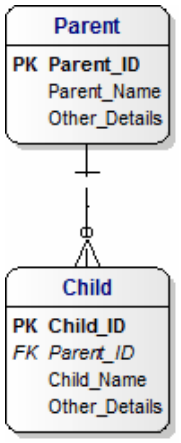


3.4.2 The Relationship showing Keys

This Data Model produced by a Data Modelling Tool is a slightly more complex version of the same Model which shows the Primary Keys (PK) and the Foreign Keys(FK) and helps us to formulate the Relationship Test for SQL.

The Primary Key stores a unique number that identifies every record uniquely.

The Foreign Key is a Primary Key in another Table to specify a cross-reference Relationship.



In English we would say, for example :-

“Every Child must have a Parent”.

In SQL, we would say

“SELECT Child_ID, Child_Name FROM Child

WHERE Parent_ID NOT IN (SELECT Parent_ID FROM Parents)”

3.4.3 Test Templates

These Templates for Test Conditions and Results can be stored in Spreadsheet or Database Tables.

The Child Table :-

Table	Lvl	No	Validation Test	Validation Test in SQL
Child	1	L1.1	Every Child must have an ID which is a unique positive integer.	SELECT COUNT(ID) WHERE Child_ID =< 0
Child	1	L1.2	Every Child must have a unique Name	SELECT COUNT(Name) HAVING COUNT(*)>1
Child	1	L1.3	Every Child must have a Name which is not blank and not NULL.	SELECT COUNT(ID) WHERE Child_Name <>'' AND NOT NULL
Child	2	N/A		
Child	3	L3.1	Every Child must have a Parent – ie every Parent ID in the Child Table is a valid ID in Parent Table	SELECT Parent_ID WHERE NOT IN(SELECT ID FROM Parent)
Child	4	N/A		
Child	5	N/A		

The Parent Table :-

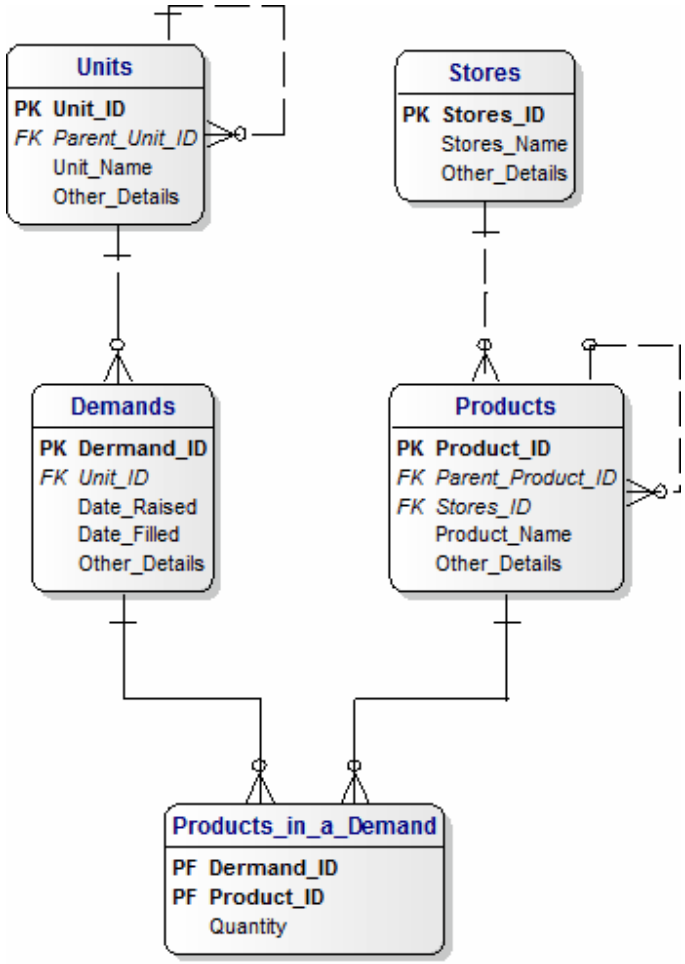
Table	Lvl	No	Validation Test	Validation Test in SQL
Parent	1	L1.1	Every Parent must have an ID which is a unique positive integer.	SELECT COUNT(ID) WHERE Parent_ID =< 0
Parent	1	L1.2	Every Parent must have a unique Name	SELECT COUNT(Name) HAVING COUNT(*)>1
Parent	1	L1.3	Every Parent must have a Name which is not blank and not NULL.	SELECT ID WHERE Parent_Name <>'' AND NOT NULL
Parent	2	N/A		
Parent	3	N/A		
Parent	4	N/A		
Parent	5	N/A		

Typical Results might look like this :-

Tables	Lvl	No	Condition	Validation Test	Valid Record Count	Invalid Record Count (%)	Comment
Parent and Child	3	L1.1	Every Child has a Parent	Every Parent ID in the Child Table is a valid ID in the Parent Table.	10,000	0 (100%)	OK

3.5 The Diagram showing Organizational Units, Products and Requisitions

This is a very simple but realistic diagram.



We will start by defining Tests at the top-left and proceed diagonally to the bottom right.

We will define the Tests with the Tables in alphabetic order to help in managing the Tests.

In English we would say

"Every Child must have a Parent".

In SQL, we would say

"SELECT Child_ID, Child_Name FROM Child

WHERE Parent_ID NOT IN(SELECT Parent_ID FROM Parents)"

1) For the Requisitions Table :

Level 1 :

1.1 The Requisition ID must be a positive integer and unique in the table.

1.2 The Date Raised must not be in the future.

1.3 The Date Filled can be either blank, NULL or a valid date subject to Test 2.1.

Level 2 :

2.1 The Date Filled must not be earlier than the Data Raised.

Level 3.

3.1 The Organizational Unit ID must be a Primary Key in the Organization Table.

Level 4. N/A

Level 5. N/A

2) For the Products Table :

Level 1 :

1.1 The Product ID must be a positive integer and unique in the table.

1.2 The Parent_Product_ID field must be either blank, NULL or match a Product ID.

1.3 The Product Name cannot be blank or NULL.

Level 2 : N/A

Level 3.

3.1 The Stores ID must be a Primary Key in the Stores Table.

Level 4. N/A

Level 5. N/A

3) For the Products_in_a_Requisition Table :

Level 1 :

1.1 The Quantity must be a positive integer.

Level 2 : N/A

Level 3.

3.1 The Requisition ID must be a Primary Key in the Requisitions Table.

3.2 The Product ID must be a Primary Key in the Products Table.

Level 4.

4.1 The Product must not be a Tornado Aicraft.

Level 5. N/A

4) For the Stores Table :

Level 1 :

1.1 The Stores ID must be a positive integer and unique in the table.

1.2 The Stores Name must not be blank or NULL and must be unique.

Level 2 : N/A

Level 3. N/A

Level 4. N/A

Level 5. N/A

5) For the Organization Table :

Level 1 :

1.1 The Organizational Unit ID must be a positive integer and unique in the table.

1.2 The Organizational Unit Name must not be blank or NULL and must be unique.

1.3 The Parent_Organizational Unit_ID field must be either blank, NULL or match a Organizational Unit ID.

Level 2 : N/A

Level 3. N/A

Level 4. N/A

Level 5. N/A

This Section contains Test Templates for all the Tables in the diagram of the Things of Interest.

1) For the Requisitions Table

Table	Lvl	No	Field	Validation Test
Requisitions	1	L1.1	Requisition_ID	Must be a positive integer and unique in the table.
Requisitions	1	L1.2	Date_Raised	Must not be in the future
Requisitions	1	L1.3	Date_Filled	Must be either blank, NULL or a valid Date
Requisitions	2	L2.1	Date_Filled and Date_Raised	The Date Filled must not be earlier than the Data Raised
Requisitions	3	L3.1	Organizational Unit_ID	Must be a Primary Key in the Organizational Units Table.
	4	N/A		
	5	N/A		

2) For the Products Table

Table	Lvl	No	Field	Validation Test
Products	1	L1.1	Product_ID	Must be a positive integer and unique in the table.
Products	1	L1.2	Product_Name	Must not be blank or NULL
	2	N/A		
Products	3	L3.1	Stores_ID	Must be a Primary Key in the Stores Table.
	4	N/A		
	5	N/A		

3) For the Products_in_a_RequisitionTable

Lvl	No	Table	Field	Validation Test
1	L1.1	Products_in_a_Requisition	Products_in_a_Requisition_ID	Must be a positive integer and unique in the table.
1	L1.2	Products_in_a_Requisition	Quantity	Must be a positive integer
2	N/A			
3	L3.1	Products_in_a_Requisition	Requisition_ID	Must be a valid Requisition_ID in the Requisitions Table.
3	L3.2	Products_in_a_Requisition	Product_ID	Must be a valid Product_ID in the Products Table.
4	L4.1	Products_in_a_Requisition	Product_ID	Tornado aircraft cannot be requested in a Requisition. Therefore the Product must not be a Tornado Aircraft (says Justin York).
5	N/A			

4) For the Stores Table

Lvl	No	Table	Field	Validation Test
1	L1.1	Stores	Stores_ID	Must be a positive integer and unique in the table.
1	L1.2	Stores	Stores_Name	Must not be blank or NULL and must be unique
2	N/A			
3	N/A			
4	N/A			
5	N/A			

5) For the Organization Table

Lvl	No	Table	Field	Validation Test
1	L1.1	Units	Parent_Organizational Unit_ID	Must be either blank, NULL or match a Organizational Unit ID
1	L1.1	Units	Organizational Unit_ID	Must be a positive integer and unique in the table.
1	L1.2	Units	Organizational Unit_Name	Must not be blank or NULL and must be unique
2	N/A			
3	N/A			
4	N/A			
5	N/A			

4 Step-by-Step Tutorial

Step 1. Check the Knowledge Database

Check for any previous Data Quality Tasks.

Step 2. Produce a Plan

Draw up a Plan that reflects the specific requirements of this Task.

Step 3. Create an entry in the Knowledge Database

Create the initial Entry in the Knowledge Database, including initial Requirements and Terms of Reference.

Step 4. Set-up the Testing Area

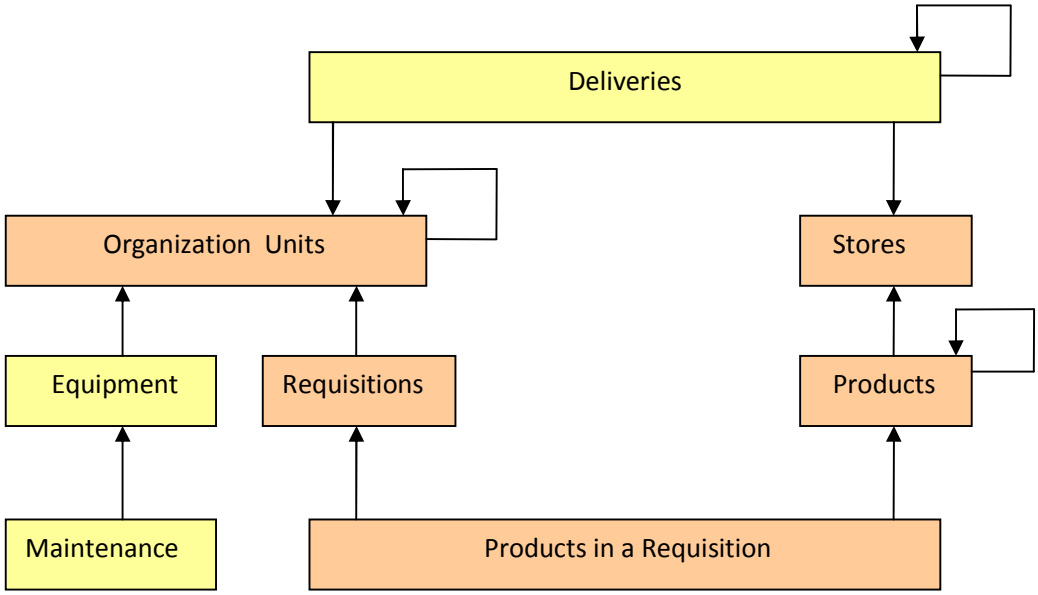
The Testing Area will provide a duplicate of the System to be tested.

If this is not possible, then a comprehensive extract must be produced so that the Test results are realistic.

Step 5. Create a Diagram with the Things of Interest

This is a Word Diagram which shows a top-level Generic view of the major Things of Interest in the scope of the Database..

This provides a common starting-point for any Data Quality Health Check, where the tables involved can be identified by starting with this Diagram.



Step 6. Obtain the Data Models

Obtain the Data Models to confirm the Data Fields and Tests for each Field.

The Case Study in Section 3 provides detailed guidance.

Table	Lvl	No	Field	Validation Test	Validation Test in SQL
	1	L1.1			
	2	L2.1			
	3	L3.1			
	4	L4.1			
	5	L5.1			

Step 7. Define the Data Validation Criteria

Go through the Diagram and use the Templates in Appendix A.1 to populate the fields and define the Tests to be run to apply the Tests.

Step 8. Run the Tests to Apply Validation Criteria

This Template shows how Validation criteria are defined.

Run Date	Table	Level	No	Field	Validation Test	Results (%)	Comments
		1	L1.1				
		2	L2.1				
		3	L3.1				
		4	L4.1				
		5	L5.1				

Step 8.1 A Sample Result Template

This Template shows a representative Health Check Report.

Run Date	Table	Level	No	Field	Validation Test	Results	Comment
Sept 16/10	Requisition	1	L1.1	Requisition ID	>0	100% OK	As expected

	s						
Sept 16/10	Requisitions	1	L1.2	Date Raised	<=System Date	95% OK	Impact ?
Sept 16/10	Requisitions	2	L2.1	Date Filled	Blank or >Date Raised	98% OK	Impact ?
Sept 16/10	Requisitions	3	L3.1	Organizational Unit ID	Must be an ID in Units Table	95% OK	Impact ?
Sept 16/10	Requisitions	4	N/A				
Sept 16/10	Requisitions	5	N/A				

Step 9. Produce the draft Health Check Report

The draft Health Check report should consolidate the results of the Data Quality Tests in a way that clarifies the situation and options.

Where appropriate CSV files could be generated to load into Excel and create Pie-charts and histograms for the Report. The Report will include recommendations for methods of Data Quality Improvement.

For example :-

“Data in the Requisitions Table is of a high quality, with percentages between 95 and 100%.

However, 5% have invalid Organizational Unit ID values. It is recommended that this discrepancy should be resolved.”

Step 10. Review the draft Health Check Report

At this point, it is appropriate to discuss the Tests used in Step 4 with the Stakeholders.

The discussion should aim to clarify any Tests that need to be changed or to be more specific so that the final Health Check Report is produced with all Validation Criteria agreed to be detailed and appropriate.

The Health Check report should consolidate the results of the Data Quality Tests in a way that clarifies the situation and options.

Where appropriate CSV files could be generated to load into Excel and create Pie-charts and histograms for the Report. The Report will include recommendations for methods of Data Quality Improvement.

For example :-

“Data in the Requisitions Table is of a high quality, with percentages between 95 and 100%.

However, 5% have invalid Organizational Unit ID values. It is recommended that this discrepancy should be resolved.”

Step 11. Repeat Steps 4 to 6 as necessary

These Steps should be repeated until the Stakeholders have agreed that the results represent the reality and when the person doing the testing is content that the rigour of the approach has been maintained.

Step 12. Review the final Health Check Report

Review the Results with the Stakeholders.

If appropriate, discuss the recommendations for methods of Data Quality Improvement.

Step 13. Update the Knowledge Database

Bring the Knowledge Database up-to-date, including the Lessons Learned.

Step 14. Decide whether to proceed to a Root Cause Analysis

Discuss the Results with the Customers and decide whether to proceed to a Root Cause Analysis.

If not, the Task is concluded, otherwise proceed to the Next Step.

Step 15. Carry out a Root Cause Analysis

Carry out a Root Cause Analysis and define Remedial Action, including Data Clean-Up Procedures.

Step 16. Decide whether to proceed to Remedial Action and Data Clean-Up

Discuss the Results with the Customers and decide whether to correct data errors.

If not, the Task is concluded, otherwise proceed to the Next Step.

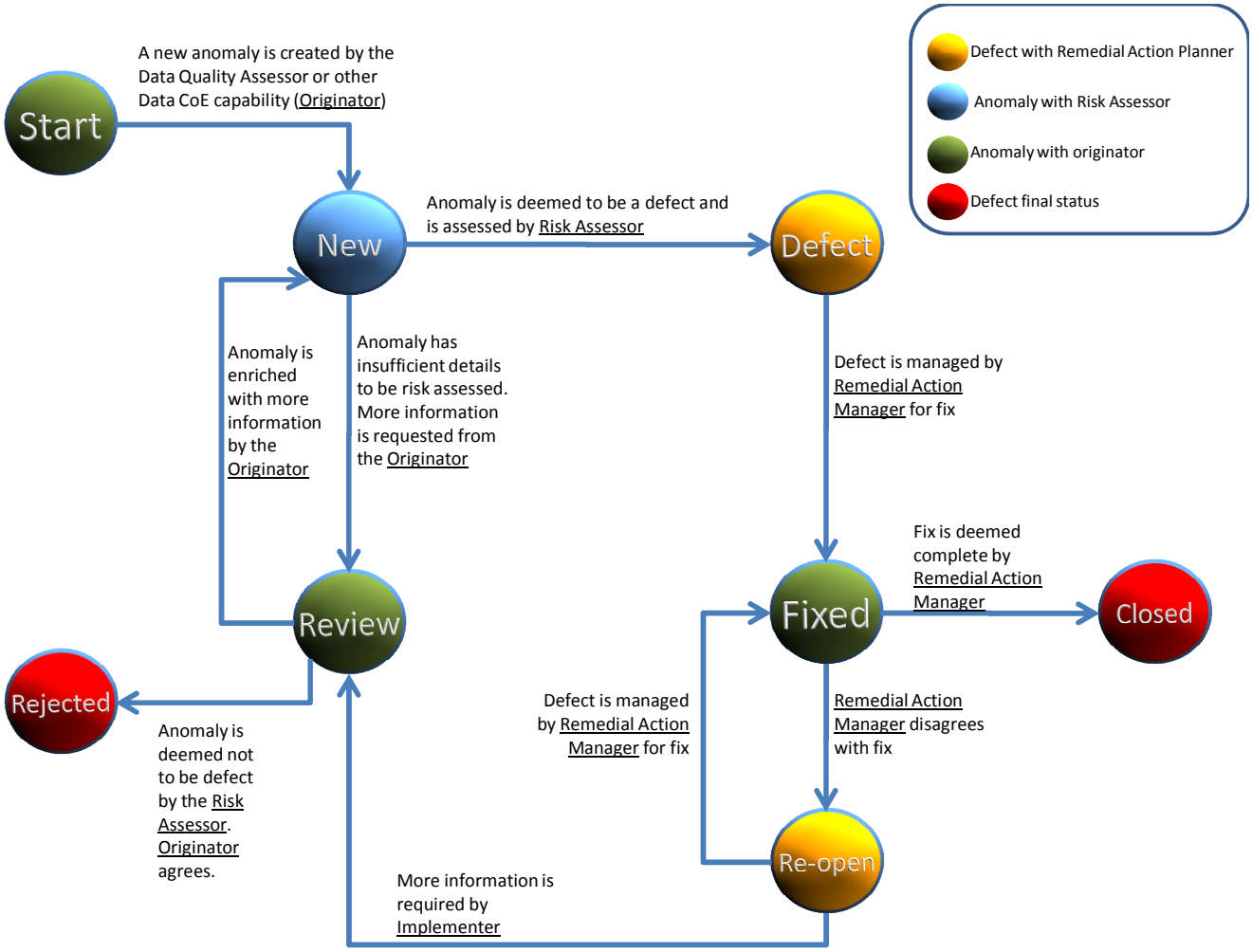
Step 17. If Task completed, update the Knowledge Database

Add details of the Task to the Knowledge Database, including the Lessons Learned.

Step 18. Run Data Clean-Up Procedures, including Remedial Action

Data Clean-up will correct errors either en route during a Data Migration or at source by taking Remedial Action .

This diagram shows details of Remedial Action :-



Step 19. Prepare Final Report for Customer

Data Clean-up will correct errors either en route during a Data Migration or at source by taking

Step 20. Update the Knowledge Database

Add details of the Task to the Knowledge Database, including the Lessons Learned.

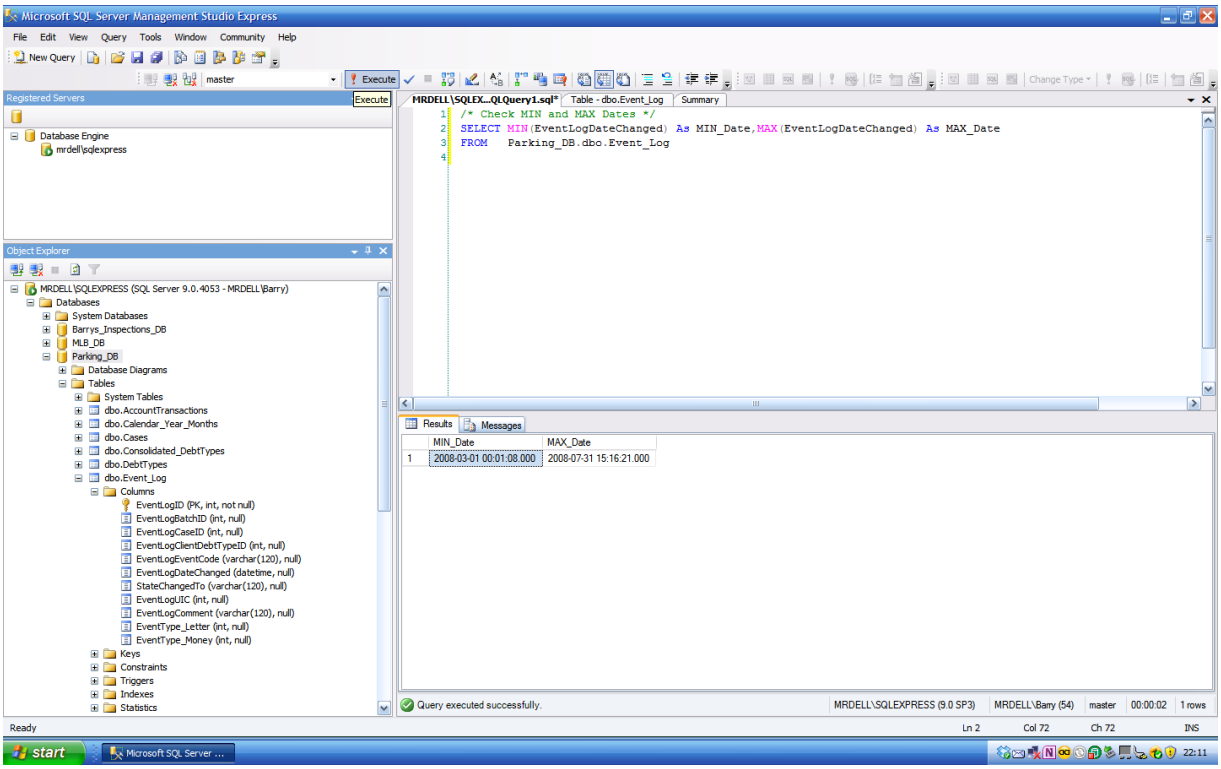
Step 21. Complete the Task

Notify the Planning Manager that that Task has been completed.

5 A Vision of the Future

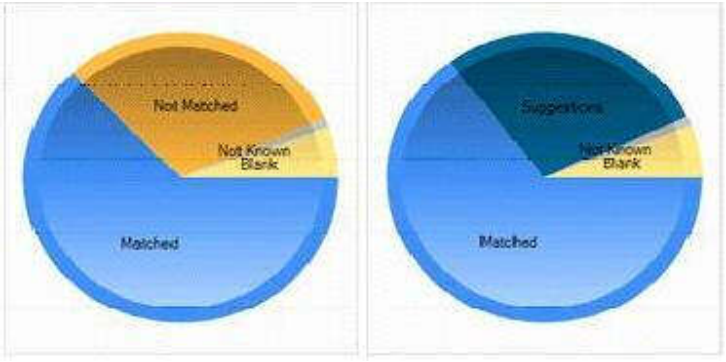
5.1 Short-Term – An Administrator's Console

This shows a simple starting-point which is provided by the Database supplier, such as Oracle, IBM or Microsoft.
It provides direct access to a Database with the ability to run SQL Scripts and save Reports.
These Reports could be used to generate CSV files to be loaded into Excel and displayed as histograms, pie-charts and so on.

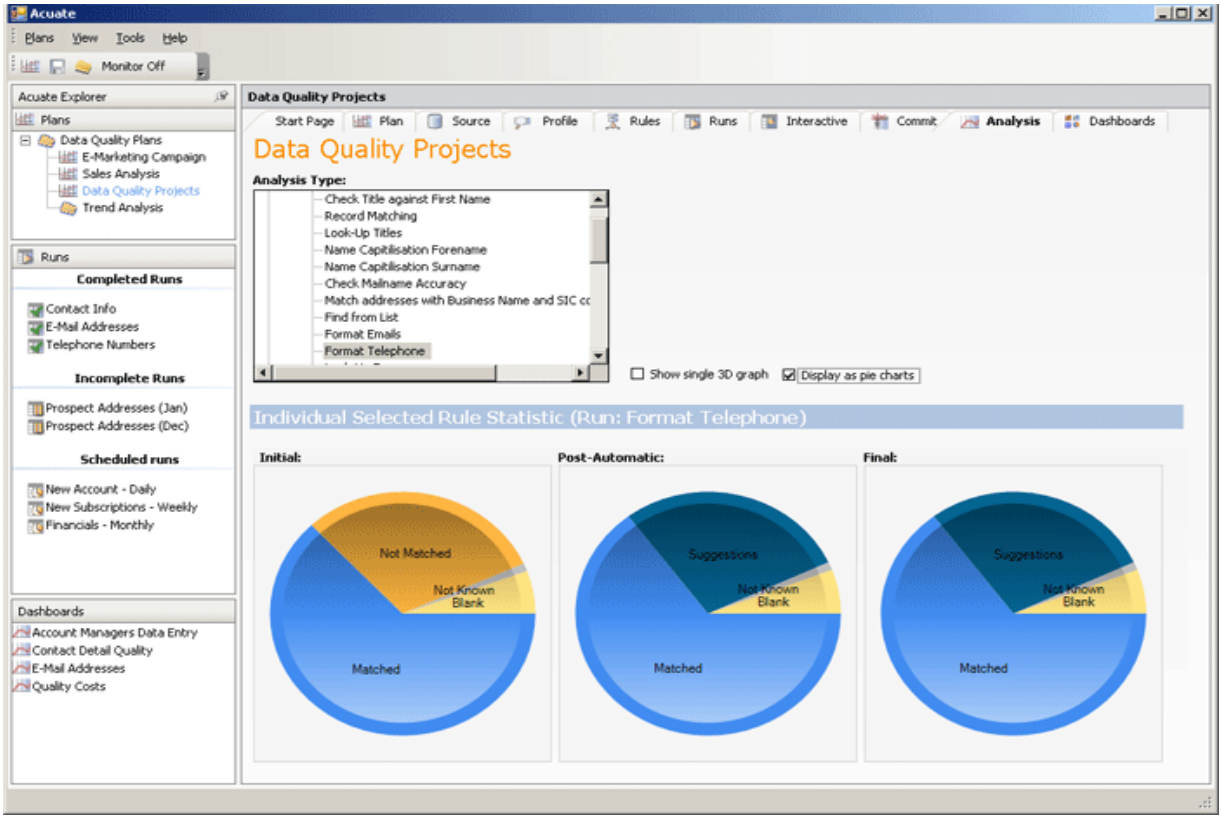


5.2 Long-Term - Dashboards

Dashboards can be very useful and can display Key Quality Indicators ('KQIs') :-



Dashboards can also show Data Quality Projects in DQ Admin Console



6 Architectures for Data Quality

6.1 Simple and Complex Architecture

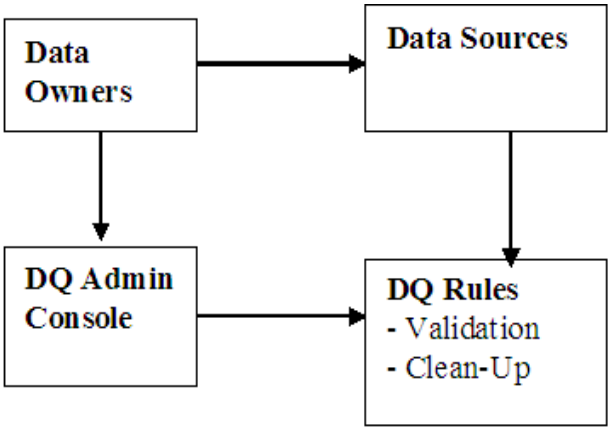
Although measuring and improving Data Quality is a simple process the ways in which it can be implemented now and in the future can be very complex.

This is particularly true for data in remote locations, such as in the 'Clouds', where it is not easy to integrate the data.

6.2 A Basic Data Quality Architecture

This consists of simple Rules coded in SQL to provide a specific solution meeting local requirements.

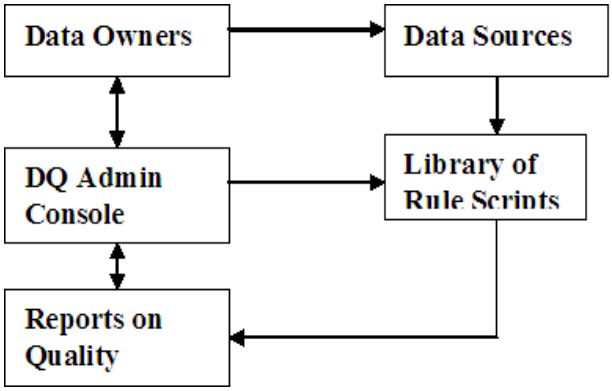
This could be provided using a simple (and free) Database Administrator's Console of the form shown in the Screenshot in Section 4.1



6.3 Intermediate Architecture

At this Stage, we add a Library of Scripts and produce standard Reports on DQ.

This might be possible using a Database Administrator's Console.



6.4 Future DQ Architecture using the Internet

This Architecture is based on Web-Services and a Service-Oriented Architecture ('SOA')

The DQ Administrator uses a Console to run DQ tests against Data Sources locally or remotely.

The Reports will be available online.

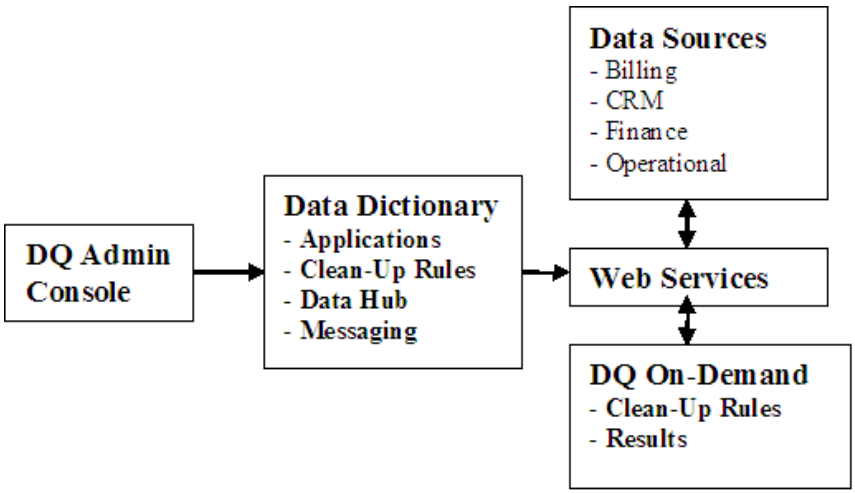
Products could be validated against the NATO Codification System which is available over the Internet and might be available by subscription using Web Services directly from the DQ Administrator's Console.

Here is the Wikipedia Link to provide a starting-point for more detailed analysis :-

- http://en.wikipedia.org/wiki/NATO_Codification_System

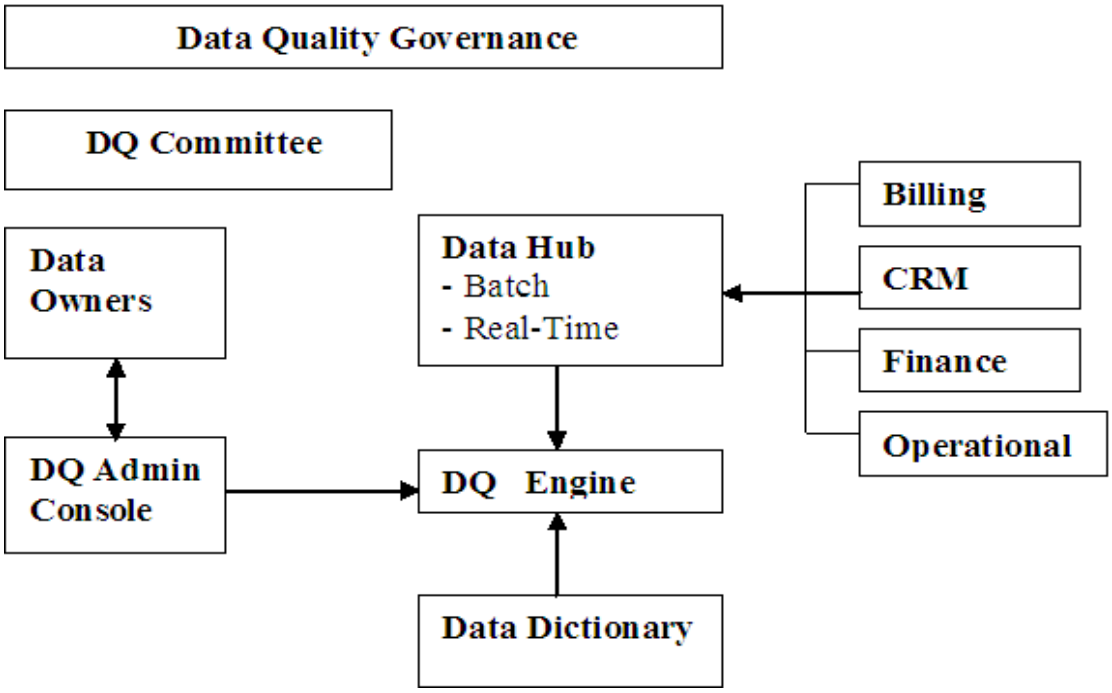
Here's the home page for the UK National Codification Bureau :-

- http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/EquipmentandLogistics/UKNCB/



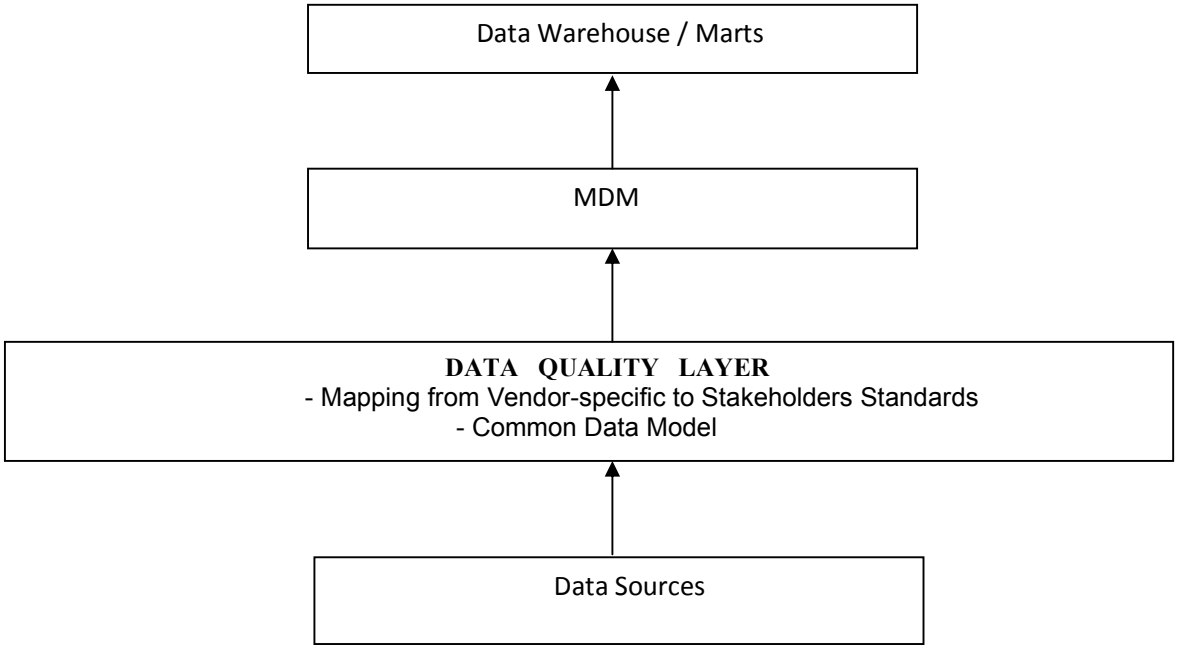
6.5 Architecture and DQ Governance

This shows DQ Committee which will exercise control over DQ and its improvement.



6.6 An Architecture for Data Quality in Migration

This shows how a Data Quality Layer can provide a Staging Area where data can be cleaned-up and transformed as it is migrated to its destination.

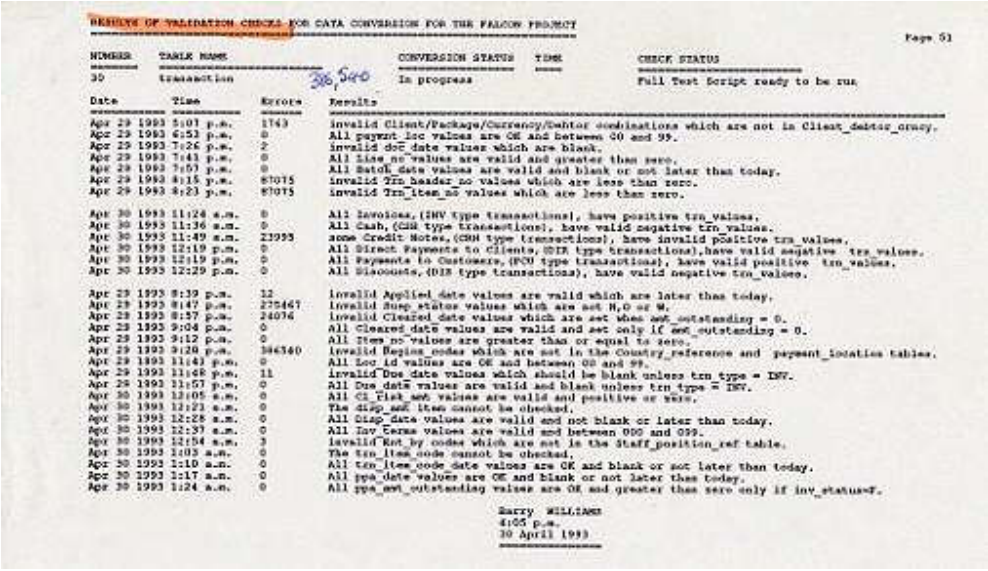


7 Screenshot for a Test Run

This shows the results of a batch Test Run with :-

- Date
- Table
- Count of errors or zero.
- Description of the Test

When this Report shows zero for all Record error counts then the Data Quality is 100%.



RESULTS OF VALIDATION CHECKS FOR DATA CONVERSION FOR THE FALCON PROJECT

Page 51

NUMBER	TABLE NAME	CONVERSION STATUS	TIME	CHECK STATUS
30	transaction	In progress		Full Test Script ready to be run

Date	Time	Errors	Results
Apr 29 1993	5:01 p.m.	1743	invalid Client/Package/Currency/Debtor combinations which are not in Client_debtor_group.
Apr 29 1993	6:53 p.m.	0	All paymt_loc values are OK and between 00 and 99.
Apr 29 1993	7:26 p.m.	2	invalid doc_date values which are blank.
Apr 29 1993	7:41 p.m.	0	All line_no values are valid and greater than zero.
Apr 29 1993	7:53 p.m.	0	All batch_date values are valid and blank or not later than today.
Apr 29 1993	8:15 p.m.	87075	invalid trn_header_no values which are less than zero.
Apr 29 1993	8:23 p.m.	87075	invalid trn_item_no values which are less than zero.
Apr 30 1993	11:24 a.m.	0	All Invoices, (INV type transactions), have positive trn_values.
Apr 30 1993	11:36 a.m.	0	All Cash, (CSH type transactions), have valid negative trn_values.
Apr 30 1993	11:49 a.m.	21993	some Credit Notes, (CRN type transactions), have invalid positive trn_values.
Apr 30 1993	12:19 p.m.	0	All Direct Payments to Clients, (DIR type transactions),have valid negative trn values.
Apr 30 1993	12:19 p.m.	0	All Payments to Customers, (PCU type transactions), have valid positive trn_values.
Apr 30 1993	12:29 p.m.	0	All Discounts, (DIS type transactions), have valid negative trn_values.
Apr 29 1993	8:39 p.m.	12	invalid Applied_date values are valid which are later than today.
Apr 29 1993	8:47 p.m.	275467	invalid Susp_status values which are not H,O or W.
Apr 29 1993	8:57 p.m.	24076	invalid Cleared_date values which are set when amt_outstanding = 0.
Apr 29 1993	9:04 p.m.	0	All Cleared_date values are valid and set only if amt_outstanding = 0.
Apr 29 1993	9:12 p.m.	0	All Item no values are greater than or equal to zero.
Apr 29 1993	9:20 p.m.	396340	invalid Region_codes which are not in the Country_reference and payment_location tables.
Apr 29 1993	11:43 p.m.	0	All Loc_id values are OK and between 00 and 99.
Apr 29 1993	11:48 p.m.	11	invalid Due_date values which should be blank unless trn_type = INV.
Apr 29 1993	11:57 p.m.	0	All Due_date values are valid and blank unless trn_type = INV.
Apr 30 1993	12:05 a.m.	0	All CL_Risk_amt values are valid and positive or zero.
Apr 30 1993	12:23 a.m.	0	The disp_amt item cannot be checked.
Apr 30 1993	12:28 a.m.	0	All Disp_date values are valid and not blank or later than today.
Apr 30 1993	12:37 a.m.	0	All Inv_terms values are valid and between 000 and 099.
Apr 30 1993	12:54 a.m.	3	invalid Ent_by codes which are not in the Staff_position_ref table.
Apr 30 1993	1:03 a.m.	0	The trn_item_code cannot be checked.
Apr 30 1993	1:10 a.m.	0	All trn_item_code_date values are OK and blank or not later than today.
Apr 30 1993	1:17 a.m.	0	All ppa_date values are OK and blank or not later than today.
Apr 30 1993	1:24 a.m.	0	All ppa_amt_outstanding values are OK and greater than zero only if inv_status=F.

Barry WILLIAMS
4:05 p.m.
30 April 1993

Appendix A. Templates

Appendix A.1 Data Quality Tests

Validation criteria are defined in unambiguous English so that they can be signed-off by the Data Owner and implemented in SQL.

They are agreed with the User or Subject-Matter Expert (SME) and form the basis for the determination of the Quality of the data in the Database.

This Template shows how Validation criteria are defined.

Run Date	Level	No	Table	Field	Validation Test	Results (%)	Comments
	1	L1.1					
	2	L2.1					
	3	L3.1					
	4	L4.1					
	5	L5.1					

Appendix A.2 Data Quality Improvement

Clean-up criteria are defined in English and implemented in SQL as shown in this Template.

Run Date	Level	No	Table	Field	Clean-Up Rules	Results	Comments
	1	L1.1					
	2	L2.1					
	3	L3.1					
	4	L4.1					
	5	L5.1					