Previews of course books are provided as an opportunity to see the quality of material used at TDWI educational events. The selected pages are representative of the entire course book. These pages cannot be printed.

The previews are intended to help you select the courses that best fit your needs.
Data Quality Assessment—Practical Skills
INTRODUCTION

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Quality Assessment Overview</td>
<td>I- 1</td>
</tr>
<tr>
<td>Project Steps</td>
<td>I-17</td>
</tr>
<tr>
<td>Course Objective</td>
<td>I-33</td>
</tr>
</tbody>
</table>
Data quality assessment is the cornerstone of any data quality program. The objective of the data quality assessment is to identify data errors and measure their impact on various business processes.

Without monitoring even the best preventive measures might eventually begin to fail without us noticing data quality deterioration until it is too late. To monitor data quality we must perform data quality assessment recurrently, compare the results, and observe the dynamics.
Introduction

Data Quality Assessment Overview

What and Why

OVERVIEW

The first step in any data quality management program is to assess magnitude of the problem. It is accomplished through the process called data quality assessment. The objective of the data quality assessment is to identify data errors and measure their impact on various business processes. The results of data quality assessment can be used to correct existing data problems, improve data collection processes and prevent future data errors.

Once we have gone through all the above steps there remains just one more important aspect of a comprehensive data quality program – ongoing monitoring. Without monitoring even the best preventive measures might eventually begin to fail without us noticing data quality deterioration until it is too late. Also even perfect prevention measures will not completely eliminate new data problems.

To monitor data quality we must perform data quality assessment recurrently, compare the results, and observe the dynamics. The results can then be used to perform more data cleansing, improve preventive measures, and sharpen our understanding of data imperfections.
Data Quality Assessment Overview

Project Team

IT Group

Data Quality Group / Data Stewardship Group

Business Users
Data Quality Assessment Overview

Project Team

OVERVIEW

Data quality assessment projects tend to follow one of two scenarios. In the first scenario they fall into the laps of technical people within the IT group because data quality assessment involves writing queries, manipulating data, and understanding databases. In the second scenario data quality assessment is performed inside business units by the data users, who can tell good data from bad and are mostly in need of quality data. Both of these scenarios are flawed.

A good data quality assessment team must include both, IT specialists and business users, ideally at least two of each kind. In addition it needs data quality experts – those who have the firsthand experience in designing, implementing, and fine-tuning data quality rules.
# Data Quality Assessment Overview

## Project Plan

<table>
<thead>
<tr>
<th>Activity</th>
<th>IT Group</th>
<th>DQ Group</th>
<th>Business Group</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define project scope and objectives</td>
<td>Planning</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Gather data and metadata</td>
<td>Preparation</td>
<td>100%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>Design and implement data quality rules</td>
<td>Implementation</td>
<td>50%</td>
<td>100%</td>
<td>25%</td>
</tr>
<tr>
<td>Fine-tune data quality rules and build data quality scorecard</td>
<td>Fine-Tuning</td>
<td>10%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

The project plan is divided into five phases: Planning, Preparation, Implementation, Fine-Tuning, and Time. The IT Group, DQ Group, and Business Group are responsible for different aspects of each phase. The table shows the percentage of responsibility and the duration for each activity.
Data Quality Assessment Overview

Project Plan

OVERVIEW

Data quality assessment projects consist of four phases: planning, preparation, implementation, and fine-tuning. The recommended project team for an average size data quality assessment project (e.g. assessment of an HR database) is made of 2 data quality experts, 1 or 2 IT professionals, and at least 2 business users (data experts).

Planning phase is usually the shortest, except when data quality assessment is planned as an enterprise-wide initiative. Preparation phase also usually does not take more than a couple of weeks unless of course we are dealing with a legacy database. In that case the length can easily double. Implementation and fine-tuning phases take the bulk of the time, about 50% each. Usually the implementation phase has a more predictable timeline, while fine-tuning can stretch and must be better controlled.

Data quality experts will be busy throughout the project, while IT and business groups will only be involved about half of the time. Also, IT group’s services are mostly required during preparation and implementation phases, while the business users are heavily taxed during fine-tuning.
## MODULE ONE
### DATA QUALITY RULES

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Domain Constraints</td>
<td>1-1</td>
</tr>
<tr>
<td>Relational Integrity Rules</td>
<td>1-25</td>
</tr>
<tr>
<td>Attribute Dependency Rules</td>
<td>1-39</td>
</tr>
<tr>
<td>Rules Using Secondary Data Sources</td>
<td>1-71</td>
</tr>
<tr>
<td>Rules for Historical Data</td>
<td>1-85</td>
</tr>
<tr>
<td>Rules for State-Dependent Objects</td>
<td>1-121</td>
</tr>
</tbody>
</table>
Attribute Domain Constraints

Introduction

- Invalid SSN Format
- BirthDate out of range
- Invalid Gender code
- Missing LastName
Attribute Domain Constraints

Introduction

DEFINITION At the most atomic level the data in any database consists of individual values of various attributes. Those values represent measurements of the characteristics of real world people, things, places, or events. Since real world objects cannot take any shape and form, values of individual attributes must also stay within certain limits.

EXAMPLE My personal record in the table has invalid ‘SSN’ format, missing ‘LastName’, invalid ‘Gender’ code, and out of range ‘BirthDate’. All these values violate attribute domain constraints.
Relational Integrity Rules

Rule Types

Relational integrity rules are derived from the analysis of relational data models.
Relational Integrity Rules

Rule Types

DEFINITIONS

Relational integrity rules are derived from the analysis of relational data models. These rules are relatively easy to identify and implement, which makes relational data model a starting point in any data quality assessment project.

Identity rules make sure that every record in a database table corresponds to one and only one real world entity and that no two records reference the same entity.

Reference rules ensure that every reference made from one entity occurrence to another entity occurrence can be successfully resolved.

Cardinal rules define constraints on the allowed number of related occurrences between entities.

Inheritance rules express integrity constraints on entities that are associated through generalization and specialization.
Rules for Historical Data

Event Dependencies

Various events in the event histories often affect the same objects. Because of that different events may be interdependent. Data quality rules can use these dependencies to validate the event histories.

The simplest event dependency restricts frequency of the events. The most complex type of rules applies to situations when events are tied by a cause-and-effect relation.

Patients may be expected to visit the dentist at least every 6 months for the regular checkups.

Mounting a dental crown will involve several visits. The nature, spacing, and duration of the visits are related.
Rules for Historical Data

Event Dependencies

OVERVIEW

Various events in the event histories often affect the same objects. Because of that different events may be interdependent. Data quality rules can use these dependencies to validate the event histories. In most cases event dependencies can only be found by extensive analysis of the nature of events.

The simplest event dependency restricts frequency of the events. The most complex type of rules applies to situations when events are tied by a cause-and-effect relation.

EXAMPLE

An airplane is required to undergo extensive maintenance after a certain number of flights.

Patients may be expected to visit the dentist at least every 6 months for the regular checkups. Mounting a dental crown will involve several visits. The nature, spacing, and duration of the visits are related.
Rules for State-Dependent Objects

Transition Constraints

*State-transition constraints* limit state changes to those allowed by the state-transition model.

*State-action constraints* require that each action is consistent with the change in the object state.

Transition from state ‘Active’ to state ‘On Leave’ is erroneously accompanied by ‘RESIGN’ action. Only ‘LOA’ action is allowed for this transition.

Transition from state ‘Terminated’ to state ‘On Leave’ is not allowed.
Rules for State-Dependent Objects

Transition Constraints

**DEFINITION**

*State-transition constraints* limit state changes to those allowed by the state-transition model. State-transition constraints are often represented by the state-transition matrix, with “from” states listed in rows and “to” states listed in columns. Invalid transitions are marked with ‘X’ at the intersection.

*State-action constraints* require that each action is consistent with the change in the object state.

**EXAMPLE**

Transition from ‘Terminated’ (T) to ‘On Leave’ (L) is not allowed by the state-transition model and is therefore invalid.

Transition from ‘Active’ (A) to ‘On Leave’ (L) is erroneously accompanied by ‘RESIGN’ action. Only ‘LOA’ action is allowed for this transition.
## MODULE TWO
IMPLEMENTING DATA QUALITY RULES

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designing Data Quality Rules</td>
<td>2-1</td>
</tr>
<tr>
<td>Building Error Catalogue</td>
<td>2-11</td>
</tr>
<tr>
<td>Fine-Tuning Data Quality Rules</td>
<td>2-31</td>
</tr>
</tbody>
</table>
Designing Data Quality Rules

Defining Project Scope

Data quality is measured by its fitness to the purpose of use.

- Tax Reporting
  - All employees receiving taxable compensation.
  - Compensation data required on annual basis.
  - Compensation data required for the current and previous years.
  - Compensation amounts must be accurate to the nearest $1.

- Calculating Retirement Benefits
  - All past and current employees eligible for retirement benefits.
  - Compensation data is required on monthly basis.
  - Compensation data is required for the last 5 years.
  - Compensation amounts must be accurate to the nearest $100.

Define Target Subject Populations
Define Target Entities and Attributes
Define Target Record Subsets
Describe Data Usage and Data Requirements

What a beautiful castle, dear!
We must charge!
The walls are in ruins!
Designing Data Quality Rules

Defining Project Scope

OVERVIEW
Data quality is measured by its fitness to the purpose of use. The first step in data quality rule design is to determine how the data is used and what are the quality requirements. Steps involved in defining project scope include:

- Define target subject populations
- Define target entities and attributes
- Define target record subsets
- Describe data uses and data requirements

EXAMPLE
For the purpose of tax reporting only compensation data for the employees receiving taxable compensation are relevant. On the other hand, in order to calculate retirement benefits we need data for all current and past employees eligible for the benefits. Also, only records for the last 2 years may be required for tax reporting, while amounts for the last 5 years might be used in retirement calculations.

Employee compensation amounts used for tax reporting must be accurate to the nearest $1. The same compensation amounts used to calculate retirement benefits must only be precise to the nearest $100. Further, annual compensation will suffice for tax reporting, while monthly amounts are used in benefit calculation. Thus, if January amount is short by $500, while February amount is $500 too high, the data is still perfectly fine for tax purposes, but inaccurate for benefit calculation.
Building Error Catalogue

Error Cataloguing

**ERROR** table contains list of all errors. Each error has unique ErrorID.

**ERROR_RECORD** table contains references to actual erroneous records in data tables.

Rule 31 E_EMPLOYEE_PROFILE_GENDER finds an error and logs it into ERROR table with ErrorID=7470. Error details are listed in the Message field.

A linked row is entered into the ERROR_RECORD table identifying the erroneous record as coming from E_EMPLOYEE_PROFILE table with RecordID=13. The trace is complete!
### Building Error Catalogue

#### Error Cataloguing

**ERROR CATALOGUE**

*Error Catalogue* is a set of entities, which collectively store information about all identified data errors. Error catalogue must reference data quality rule that found each error as well as data records it affected.

**ERROR ENTITY**

*ERROR* entity is the first part of the error catalogue. It simply lists all errors and has 3 attributes:
- **ErrorID** is a unique identifier for each error.
- **RuleID** references data quality rule that found the error.
- **Message** provides additional error details.

**ERROR_RECORD ENTITY**

*ERROR_RECORD* entity is the second part of the error catalogue. It contains references to actual erroneous records in data tables and has 3 attributes:
- **ErrorID** is the identifier referencing the error.
- **TableName** references data table where error is found.
- **RecordID** references the erroneous record itself.

**EXAMPLE**

Rule #31 ‘E_EMPLOYEE_PROFILE_GENDER’ finds a new error and logs it into *ERROR* table with *ErrorID*=7470.

A linked row is entered into the *ERROR_RECORD* table identifying the erroneous record as coming from *E_EMPLOYEE_PROFILE* table with *RecordID*=13.
# MODULE THREE

BUILDING DATA QUALITY SCORECARD

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate Scores Overview</td>
<td>3-1</td>
</tr>
<tr>
<td>Score Tabulation</td>
<td>3-11</td>
</tr>
<tr>
<td>Data Quality Scorecard</td>
<td>3-25</td>
</tr>
<tr>
<td>Recurrent Data Quality Assessment</td>
<td>3-39</td>
</tr>
</tbody>
</table>
Aggregate Scores Overview

Introduction

*Aggregate scores* provide high-level estimates of the data quality. Each score aggregates errors identified by the data quality rules into a single number – a percentage of good data records among all target data records.

8% of records for the last 2 years are incorrect

11% of records that came from payroll system are bad
Aggregate Scores Overview

Introduction

**OVERVIEW**

*Aggregated scores* provide high-level estimates of the data quality. Each score aggregates errors identified by the data quality rules into a single number – a percentage of good data records among all data records.

For each score a subset of relevant records, fields, and data quality rules is selected. Only the selected records are considered and only errors identified by the selected data quality rules in the selected records are counted.

Many scores can be created for a single database. They can measure data fitness for various purposes, indicate quality of various data collection processes, etc.

Aggregated scores help make sense out of the numerous error reports produced by data quality rules. Analysis of aggregated scores answers key data quality questions:

- What is the impact of the errors in your database?
- What are the sources of the errors?
- Where can most errors be found?
Score Tabulation
Record-Level Score Formulas

Completeness Score  = \frac{\text{Count of All Relevant Records}}{\text{Count of All Relevant Records} + \text{Count of Missing Records}}

Accuracy Score  = \frac{\text{Count of All Relevant Records} - \text{Count of Erroneous Records}}{\text{Count of All Relevant Records}}

Overall Score  = \frac{\text{Count of All Relevant Records} - \text{Count of Erroneous Records}}{\text{Count of All Relevant Records} + \text{Count of Missing Records}}
Score Tabulation

Record-Level Score Formulas

OVERVIEW

The Overall Score is defined as the fraction of “good” records among all records. We can count “good” records by simply eliminating erroneous ones from all relevant records. Further, to get the count of all records we start with existing relevant records and add missing records, that otherwise would have been relevant.

Sometimes it is more convenient to think about missing and erroneous records separately. In order to satisfy this view we break the Overall Score into two components – Completeness Score and Accuracy Score. The product of these two scores equals the Overall Score.

The first step is to find and count all relevant records. With explicit queryable conditions for each table the counting is rather simple. However simple conditions are relatively uncommon. A more practical solution is to somehow mark all relevant records or create a separate list.

The next step is to count data records with possible errors. These are the records among identified relevant records that violated at least one of the selected data quality rules. We also need to count missing records. Both tasks can be accomplished by querying the error catalogue.
Data quality scorecard is the central product of the data quality assessment project. It provides comprehensive information about data quality and allows both aggregated analysis and detailed drill-downs.
Data Quality Scorecard

Introduction

OVERVIEW

Data quality scorecard is the central product of the data quality assessment project. It provides comprehensive information about data quality and allows both aggregated analysis and detailed drill-downs. Well-designed data quality scorecard is the key to understanding how well the data supports various data-driven projects. It is also critical for making good decisions about data quality initiatives.

Data quality scorecard as an information pyramid. At the top level are aggregate scores; at the bottom level is information about data quality of individual data records. In the middle are various score decompositions and error reports allowing to analyze and summarize data quality across various dimensions and for different objectives.
Recurrent Data Quality Assessment

Introduction

In theory recurrent data quality assessment involves simple periodical re-running data quality rules against new data, building new data quality scorecard, and comparing the results over time.
Recurrent Data Quality Assessment

Introduction

OVERVIEW

Let’s say we would like to check the quality of data in a particular database every six months. The first time we execute assessment exactly as was discussed thus far. We bring the data to the staging area; go through data analysis and profiling; design, implement, and fine-tune data quality rules; populate the data quality metadata warehouse; and build the data quality scorecard.

The next time we want to run the assessment all we need to do is backup the staging area and data quality metadata warehouse, wipe out the error catalogue, reload the most recent data dump back to the staging area, and rerun all data quality rules. This procedure will produce the new data quality scorecard quickly and with practically no extra work. Comparison of the old and new data quality scorecard indicates changes to the data quality.